Universidade Estadual de Campinas
Instituto de Computação

Leonardo de Melo João

A framework for iterative saliency estimation on multiple image domains

Um arcabouço para estimativa de saliência em múltiplas iterações em diferentes domínios de imagem

CAMPINAS
2020

Leonardo de Melo João

A framework for iterative saliency estimation on multiple image domains

Um arcabouço para estimativa de saliência em múltiplas iterações em diferentes domínios de imagem

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Dissertation presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Computer Science.

Supervisor/Orientador: Prof. Dr. Alexandre Xavier Falcão

Este exemplar corresponde à versão final da Dissertação defendida por Leonardo de Melo João e orientada pelo Prof. Dr. Alexandre Xavier Falcão.

CAMPINAS
2020

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

J57f João, Leonardo de Melo, 1995-
A framework for iterative saliency estimation on multiple image domains / Leonardo de Melo João. – Campinas, SP : [s.n.], 2020.

Orientador: Alexandre Xavier Falcão.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Processamento de imagens. 2. Visão por computador. I. Falcão, Alexandre Xavier, 1966-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Um arcabouço para estimativa e saliência em múltiplas iterações em diferentes domínios de imagem
Palavras-chave em inglês:
Image processing
Computer vision
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Alexandre Xavier Falcão [Orientador]
Silvio Jamil Ferzoli Guimarães
Hélio Pedrini
Data de defesa: 09-10-2020
Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0003-4625-7840
- Currículo Lattes do autor: http://lattes.cnpq.br/1735436819285585

Universidade Estadual de Campinas
Instituto de Computação

Leonardo de Melo João

A framework for iterative saliency estimation on multiple image domains

Um arcabouço para estimativa de saliência em múltiplas iterações em diferentes domínios de imagem

Banca Examinadora:

- Prof. Dr. Alexandre Xavier Falcão
  Universidade Estadual de Campinas
- Prof. Dr. Silvio Jamil Ferzoli Guimarães
  Pontifícia Universidade Católica de Minas Gerais
- Dr. Helio Pedrini
  Universidade Estadual de Campinas

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 09 de outubro de 2020

Agradecimentos

I would like to greatly thank my advisor, Prof. Dr. Alexandre Xavier Falcão, for all his teaching, his support and guidance during these two years pursuing my master's degree. Thank you for constantly helping me overcome my academic challenges. Furthermore, I thank CNPq for financing this project.

I am also grateful for my family and friends who have been extremely supportive all the way, providing me the best of times to take a breath from all the work and in midst of this insane *covid* pandemic.

More importantly, I thank God for all the opportunities and help He provided. I could not have done it by myself.

Resumo

A detecção de objetos salientes estima os objetos que mais se destacam em uma imagem. Os estimadores de saliência não-supervisionados utilizam um conjunto predeterminado de suposições a respeito de como humanos percebem saliência para identificar características discriminantes de objeto salientes. Como esses métodos fixam essas suposições predeterminadas como parte integral de seu modelo, esses métodos não podem ser facilmente estendidos para cenários específicos ou outros domínios de imagens. Nós propomos, então, um arcabouço iterativo para estimação de saliência baseado em superpixels, intitulado ITSELF (Iterative Saliency Estimation fLexible Framework). Nosso arcabouço permite que o usuário adicione múltiplas suposições de saliência para melhor representar seu modelo. Graças a avanços em algoritmos de segmentação por superpixels, mapas de saliência podem ser utilizados para melhorar o delineamento de superpixels. Combinando algoritmos de superpixels baseados em informações de saliência com algoritmos de estimação de saliência baseados em superpixels, nós propomos um ciclo para auto melhoria iterativa de mapas de saliência. Nós comparamos o ITSELF com outros dois estimadores de saliência no estado-da-arte em cinco métricas e seis conjuntos de dados, dos quais quatro são compostos por imagens naturais, e dois são compostos por imagens biomédicas. Os experimentos mostram que nossa abordagem é mais robusta quando comparada aos outros métodos, apresentando resultados competitivos em imagens naturais e os superando em imagens biomédicas.

Abstract

Saliency object detection estimates the objects that most stand out in an image. The available unsupervised saliency estimators rely on a pre-determined set of assumptions of how humans perceive saliency to create discriminating features. These methods cannot be easily extended for specific settings and different image domains by fixing the pre-selected assumptions as an integral part of their models. We then propose a superpixel-based ITerative Saliency Estimation fLexible Framework (ITSELF) that allows any user-defined assumptions to be added to the model when required. Thanks to recent advancements in superpixel segmentation algorithms, saliency-maps can be used to improve superpixel delineation. By combining a saliency-based superpixel algorithm to a superpixel-based saliency estimator, we propose a novel saliency/superpixel self-improving loop to enhance saliency maps iteratively. We compare ITSELF to two state-of-the-art saliency estimators on five metrics and six datasets, four of them with natural images and two with biomedical images. Experiments show that our approach is more robust than the compared methods, presenting competitive results on natural image datasets and outperforming them on biomedical image datasets.

List of Figures

1.1 (a) Original image. (b-c) SMD [39] and DRFI [26] saliency maps. 14
1.2 (a) Original images; (b) Ground-truth segmentation; (c) ITSELF saliency map. 15
1.3 ITSELF's overview. Note the saliency-superpixel loop depicted in blue. By using an object-based superpixel segmentation algorithm ITSELF is able to iteratively enhance both the saliency estimation and superpixel segmentation. As a result, ITSELF outputs both enhanced results. 16
1.4 (a) Original image. (b) DRFI saliency map; (c) SMD saliency map. (d) ITSELF saliency map. Note how ITSELF does not highlight the immediate surroundings of salient objects and also highlighted the other chairs. 16
1.5 (a) Input image. (b) Center-surround prior; (c) Red-yellow prior; (d) White prior; (e) Global color contrast prior; (f) Combined priors. 17
1.6 (a) Original image. (b-c) From the left to the right, results and superpixel segmentation of ITSELF's iterations 1, 5 and 8, respectively. Note how the number of superpixels change to incorporate multiple scales. 18

2.1 (a) The image graph with three starting seeds (a,b, and c); (b) All trivial path are attributed the initialization cost; (c-f) the iterative execution of the IFT algorithm where the seeds conquer the vertices more strongly connected to them. 22
2.2 (a) Input image with object seeds on blue and background seeds on red. (b) The gradient map used to compute the function weights; (c-d) The segmentation result using the IFT over the $\boldsymbol{f}_{max}$ and $\boldsymbol{f}_{sum}$ functions, respectively. The red arrows point to low gradient areas between foreground and background 23
2.3 (a) Input image with object seeds on blue and background seeds on red. (b) The gradient map used to compute the function weights; (c-d) The segmentation result using the IFT over the $\boldsymbol{f}_{max}$ and $\boldsymbol{f}_{sum}$ functions, respectively 24
2.4 (a,b) Input image and ground-truth. (c) Superpixel segmentation with thirty two superpixels; (d) a possible resulting saliency map 26
2.5 (a) Input image. (b) Ground-truth; (c) Super-segmented salient object [5]; (d) Sub-segmented salient object. Both segmentation images were computed using OISF [5] with 200 superpixels and the ground-truth as the input saliency map 28

2.6 (a,e) The original image and its saliency map, respectively; (b,f) The cost maps using only color difference and combining color difference to object information, respectively; (e-f) The result of initial iterations that was not effected by the object information. (g) The resulting superpixels using only color difference; (h) The resulting superpixels combining color difference and object information. 29
2.7 (a,b) The original image and its saliency map, respectively; (c,d) OISF segmentation insuring better superpixel regularity ($\alpha = 12$, $\beta = 0.8$) and better boundary adherence ($\alpha = 0.8$, $\beta = 12$; (e,f) OISF segmentation using different object information weight on delineation ($\gamma = 0.5$ and $\gamma = 2.0$). The cyan arrows indicate regions where the object information allowed separation of low-contrast boundaries between foreground and background. 30
2.8 (a) The original automaton cells derived from a fictional image, where a hollow white square contains an inner background square (in gray); (b-d) The inner square start changing state (depicted by the pink borders) due to the state of it's adjacent object cells (in green). 31
2.9 (a) Input image. (b-c) Result of saliency map integration using $\lambda \in \{0.01, 0.1\}$, respectively. Note that although a higher λ creates more homogeneous salient objects, less salient object parts may be lost. 32

4.1 Detailed ITSELF's overview. The framework user can define specific saliency models according to prior knowledge and query selection strategies. Note that the saliency and prior map integration is depicted as a plus sign. . . . 40
4.2 (a) Input image. (b-d) The result of ITSELF on iterations one, five and eight (final) using the automaton; (e-g) The result of ITSELF on the same iterations as (b-d) but without using the automaton to take previous iterations into consideration. 43
4.3 (a) Original image. (b) Superpixel Segmentation; (c) and (d) Center prior maps with $\sigma_1 = 0.1$ and $\sigma_1 = 0.9$, respectively. 44
4.4 (a) Original image. (b) Superpixel Segmentation; (c) and (d) Global color-contrast prior maps without and with the smoothness step, respectively. Note how slight changes in tones of green impact negatively the method without smoothness. 45
4.5 (a) Original image with object scribbles in light-blue. (b) Black prior map with $\sigma_3 = 0.5$. Additionally, we reduced the saliency of black regions connected to the image boundaries because of the natural color of the xray plate. 46
4.6 (a) Original image. (b) Saliency map before multiplying by the proposed color-saliency based prior; (c) the color-saliency based prior derived from (b); (d) the result of multiplying the initial saliency with the proposed prior. Note the error reduction on background saliency. 47
4.7 (a) Original image. (b) Result of the focus prior. (c) Estimated object edges; (d) Object-based superpixel segmentation. 48
4.8 (a) Original image. (b) Object mask of the parasite. (c) Superpixel segmentation; (d) Ellipse Matching of each superpixel 50
4.9 (a) Ellipse-based prior without size filtering (b) Result of reducing region saliency by size. 51

4.10 (a) Original image with object scribbles in light-blue. (b) Scribble-based location prior map. 51
4.11 (a) Original image with object scribbles in light-blue. (b) Scribble-based location prior map. 52
4.12 (a) Input image. (b) The result combination of the boundary clusters; (c) The highest boundary-connectivity score cluster with $w_c = 0.453$; (d) A boundary cluster containing most of the object with boundary-connectivity score $w_c = 0.142$. Note that the combined saliency map is not the final result of ITSELF, rather it is the simple combination of the boundary clusters. 53

5.1 (a) A parasite egg (red arrow) and a fecal impurity (blue arrow) that shares similar characteristics to the eggs; (b) A heavily cluttered image with one parasite egg (red arrow) 55
5.2 (a) Input image. (b) Ground-truth segmentation; (c-f) ITSELF/SMD saliency maps with mean-saliency threshold segmentation boundaries depicted on green/blue, respectively. 58
5.3 (a) Original image. (b) Superpixel Segmentation; (c) Reported ITSELF result; (d) Improved result by removing the center and focus priors. 59
5.4 (a) Original image. (b) Ground-truth; (c) ITSELF saliency map. On the second image, note how there are contrasting green parts on the image background. 60
5.5 (a) Input image. (b) Ground-truth segmentation; (c-h) DRFI/ITSELF/SMD saliency maps with mean-saliency threshold segmentation boundaries depicted on red/green/blue, respectively. Note how ITSELF tend to create more accentuated contrast between the object and background, adhering to the boundaries. 61
5.6 (a) Original image. (b) Ground-truth; (c) ITSELF saliency map. ITSELF completely lost the smaller and brighter lung. 62
5.7 (a) Original image. (b) Ground-truth (red) and ITSELF (green) segmentations overlayed. Note the lighter yellow membrane segmented on the ground-truth that was lost by ITSELF. 62
5.8 (a) Original image. (b) Ground-truth; (c) ITSELF saliency map. ITSELF highlights the top right impurity instead of the parasite-egg. 62
5.9 (a) Input image. (b) Ground-truth segmentation; (c-h) DRFI/ITSELF/SMD saliency maps with mean-saliency threshold segmentation boundaries depicted on red/green/blue, respectively Note how ITSELF tend to create more accentuated contrast between the object and background, adhezring to the boundaries. 63

Contents

1 Introduction **13**
- 1.1 Motivation . . . 13
- 1.2 Objectives . . . 14
- 1.3 Contributions . . . 14
- 1.4 Document Organization . . . 17

2 Theoretical Background **19**
- 2.1 Notations and Definitions . . . 19
- 2.2 Image Foresting Transform . . . 20
- 2.3 Iterative Spanning Forest . . . 22
- 2.4 Superpixel-based Saliency Estimation . . . 25
- 2.5 Object-based Iterative Spanning Forest . . . 27
- 2.6 Cellular Automata for Saliency Map Integration . . . 29

3 Related Works **33**
- 3.1 Supervised Saliency Estimation . . . 33
- 3.2 Unsupervised Saliency Estimation . . . 34
- 3.3 Graph-based saliency estimation . . . 35
- 3.4 Superpixel segmentation using the IFT framework . . . 36
- 3.5 Superpixels for saliency estimation . . . 37
- 3.6 Conclusion . . . 37

4 *Iterative Saliency Estimation fLexible Framework* **39**
- 4.1 Object-based Superpixel Segmentation . . . 39
- 4.2 Superpixel-based Saliency Estimation . . . 41
- 4.3 Prior and Saliency Map Integration . . . 42
- 4.4 Prior Modeling . . . 42
  - 4.4.1 Center-surround prior . . . 42
  - 4.4.2 Global color uniqueness prior . . . 43
  - 4.4.3 Color-based priors . . . 44
  - 4.4.4 Saliency-based priors . . . 46
  - 4.4.5 Focus prior . . . 46
  - 4.4.6 Ellipse-matching prior . . . 48
  - 4.4.7 Scribbles based priors . . . 49
- 4.5 Query Selection . . . 50
  - 4.5.1 Border-based Query Selection . . . 51
  - 4.5.2 Saliency-based Query Selection . . . 53

5 Experiments and Results **54**
5.1 Datasets 54
5.2 Parameter tuning and Experimental setup 55
5.3 Evaluation Metrics 56
5.4 Natural-image dataset comparisons 57
5.5 Non-natural-image dataset comparisons 58
5.6 Failed Attempts and Implementation Details 60

6 Conclusion and Future work **65**

Bibliography **67**

Chapter 1

Introduction

1.1 Motivation

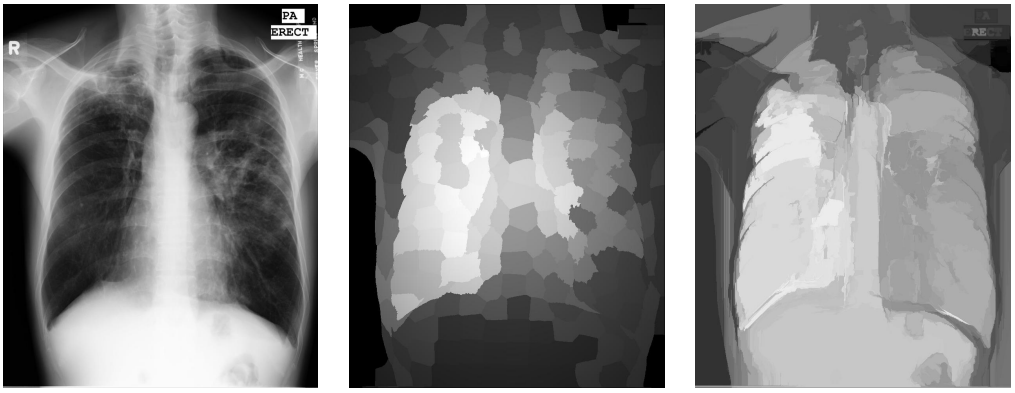
Determining visual saliency of image objects is a broadly studied subject, highly applicable to a vast number of tasks, such as image quality assessment [32], content-based image retrieval [12], and image compression [22]. Many algorithms have been proposed to estimate visual saliency, and they can be categorized as supervised and unsupervised approaches.

Supervised saliency estimators use pixel-wise ground-truth images to learn discriminant features of salient objects. The most accurate supervised algorithms are based on deep-learning techniques [50], but they require large amounts of training data annotated by humans, and the generalization of the trained models across image datasets or image domains usually requires retraining and adaptations. Unsupervised saliency estimators, however, model saliency based on some prior knowledge about the salient objects and local image characteristics, usually compromising accuracy in exchange for removing the requirement for intensive data annotation, being more flexible across image domains. In this work, we are focused on unsupervised saliency estimators.

Most unsupervised saliency estimation algorithms model saliency using a combination of bottom-up image-extracted information and top-down domain-specific assumptions. The bottom-up information often is extracted from image regions that, given modeled assumptions, have a high likelihood of being either foreground (salient) or background. These regions, namely *queries*, are compared to the rest of the image, and a similarity score defines how salient the other regions are. As example, assuming the salient objects to be centered and not contained within the limits of the image, a common strategy is to use regions in contact to the image borders as background queries. Top-down assumptions, on the other hand, use *prior* knowledge of how humans perceive saliency, *e.g.* increase the saliency of centered, focused, and vivid-colored objects.

The available methods combine a pre-selection of priors and query-defining strategies to model the saliency perception of an average viewer of natural-images. This pre-selection of assumptions allows for off-the-shelf methods that are easy to use and perform well in many scenarios. On the other hand, they are not extensible to applications that drift off of their pre-determined guesses.

For example, if we shift the image domain from natural images to medical images,



(a) (b) (c)

Figure 1.1: (a) Original image. (b-c) SMD [39] and DRFI [26] saliency maps.

the desired saliency is not modeled after the average viewer; Rather, it is modeled after a specialist's perception. For instance, say a physician is analyzing an x-ray image of the thorax: object centering and vivid colors cease to be salient object characteristics, causing the off-the-shelf methods to perform poorly. To the best of our knowledge, there is no unsupervised saliency estimation algorithm that allows the user to select or incorporate a problem-specific set of assumptions. In this regard, the state-of-the-art algorithms are not suitable for estimating saliency in multiple image domains (Figure 1.1).

1.2 Objectives

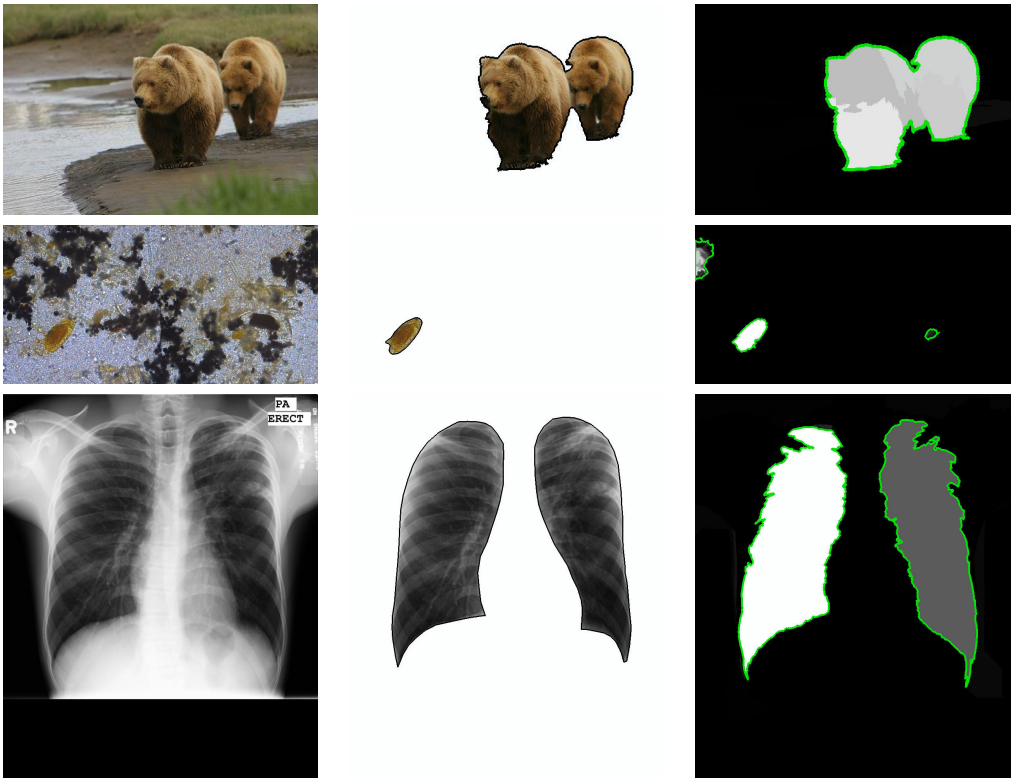
We raised two fundamental questions for our research: is it possible to build a saliency estimation method that combines multiple characteristics such that the user can customize it for a given application? If so, can the same method be applied to multiple image domains?

Therefore, the main objective of this work is to create a flexible saliency estimation framework that can easily be extended to fit the saliency model expected by the user. The proposed framework must allow any number of salient characteristics to be incorporated into the model.

1.3 Contributions

In this work, we propose the *ITerative Saliency Estimation fLexible Framework (ITSELF)*. ITSELF is a graph-based framework that allows user-defined *priors* and *query-region selection*, making it flexible to multiple image domains (Figure 1.2). Saliency is estimated by computing similarities on a superpixel graph, where the nodes are superpixels, and the arcs connect superpixels according to some adjacency relation based on the query regions. The saliency score is improved using multiple top-down prior information combined into a single prior map.

Additionally, ITSELF uses a novel approach to iteratively enhance saliency maps by using object-based superpixel delineation [5]: Saliency information is used to delineate



(a) (b) (c)

Figure 1.2: (a) Original images; (b) Ground-truth segmentation; (c) ITSELF saliency map.

better superpixels, allowing for a better saliency score to be derived from the improved superpixel segmentation. By revisiting both the superpixel-based saliency estimation with the improved superpixels, and the saliency-based superpixel segmentation with improved saliency score, we can create a virtuous cycle for enhancing saliency maps over time (Figures 1.3 1.6). This over time saliency improvement allows the creation of more intuitive saliency maps, where multiple somewhat salient objects are also detected (Figure 1.4). When compared to the existing unsupervised approaches, ITSELF can leverage the object saliency map under construction to improve the process and output a superpixel segmentation as a byproduct.

We propose new prior modeling for specific scenarios, including a shape-based, a saliency-based, and a user-drawn scribble-based prior. A fitting subset of any number of priors is merged into one final prior map using an automata-based saliency map integration step [41] (Figure 1.5). Then, the resulting prior map is combined with the bottom-up saliency map — estimated according to the selected query strategies — resulting on the output of each of ITSELF's iterations. The same saliency map integration is then used to combine the multiple results of subsequent ITSELF iterations to generate the final saliency map. By using all iteration results on the final integration, ITSELF allows for multiple scales to be considered if the number of superpixels change over time (Figure 1.6).

We compare ITSELF to two state-of-the-art unsupervised methods — namely DRFI [26] and SMD [39] — using four well-established natural-image datasets, an in-house

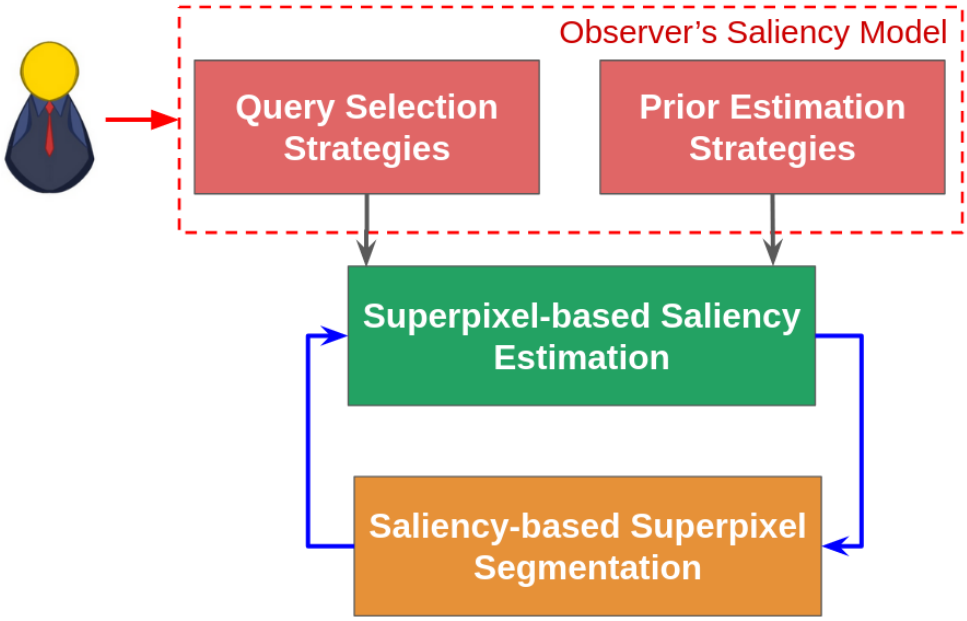


Figure 1.3: ITSELF's overview. Note the saliency-superpixel loop depicted in blue. By using an object-based superpixel segmentation algorithm ITSELF is able to iteratively enhance both the saliency estimation and superpixel segmentation. As a result, ITSELF outputs both enhanced results.

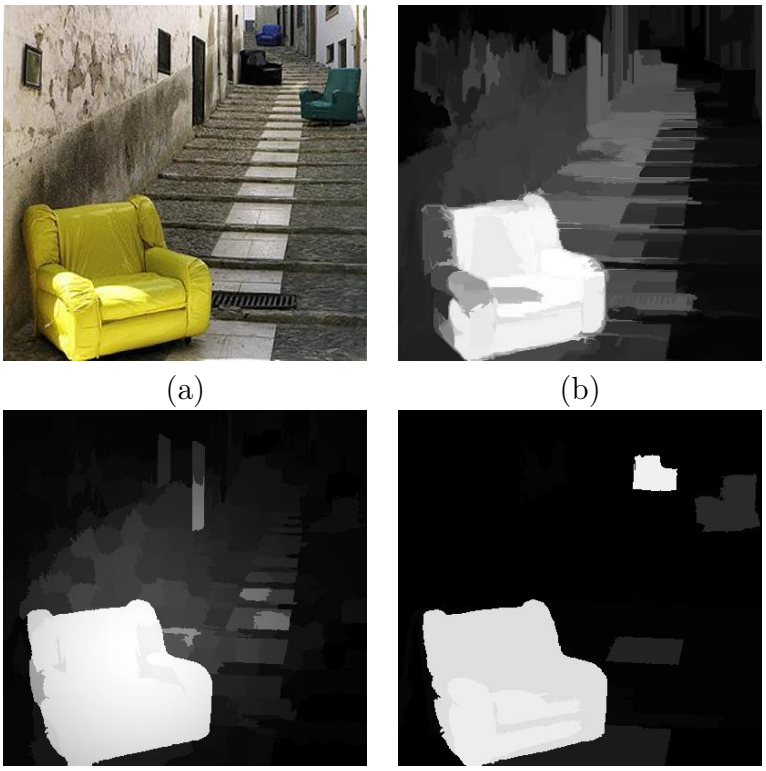


Figure 1.4: (a) Original image. (b) DRFI saliency map; (c) SMD saliency map. (d) ITSELF saliency map. Note how ITSELF does not highlight the immediate surroundings of salient objects and also highlighted the other chairs.

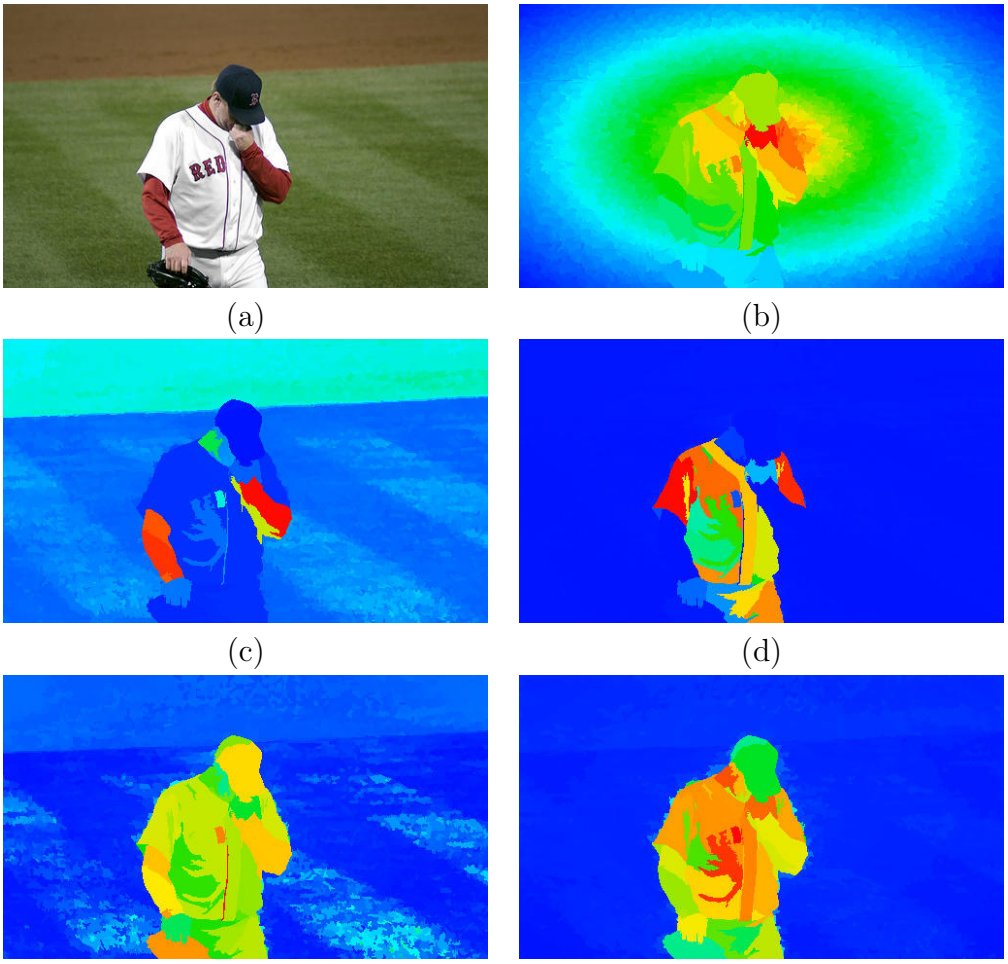


Figure 1.5: (a) Input image. (b) Center-surround prior; (c) Red-yellow prior; (d) White prior; (e) Global color contrast prior; (f) Combined priors.

biomedical image dataset of parasite-eggs, and an x-ray dataset of lung gray-scale images. Even though the selected datasets provide three different image domains, ITSELF was able to provide appropriate saliency estimations for all of them. We achieved comparable results to the state-of-the-art algorithms on natural images and considerably outperformed them on non-natural images.

Thus, the contributions of this work are: (1) a saliency estimation framework that easily allows the incorporation of domain-specific information; (2) the improvement of saliency estimation by using object-information during superpixel segmentation; (3) a novel method for iteratively enhancing both saliency maps and superpixel segmentation.

1.4 Document Organization

In this work, notations, definitions, and theoretical background are presented in Chapter 2. Later, a literature review is presented in Chapter 3, showcasing an overview of the current supervised and unsupervised saliency estimation methods, superpixel segmentation, and the usage of superpixel on saliency estimation. The core elements of ITSELF

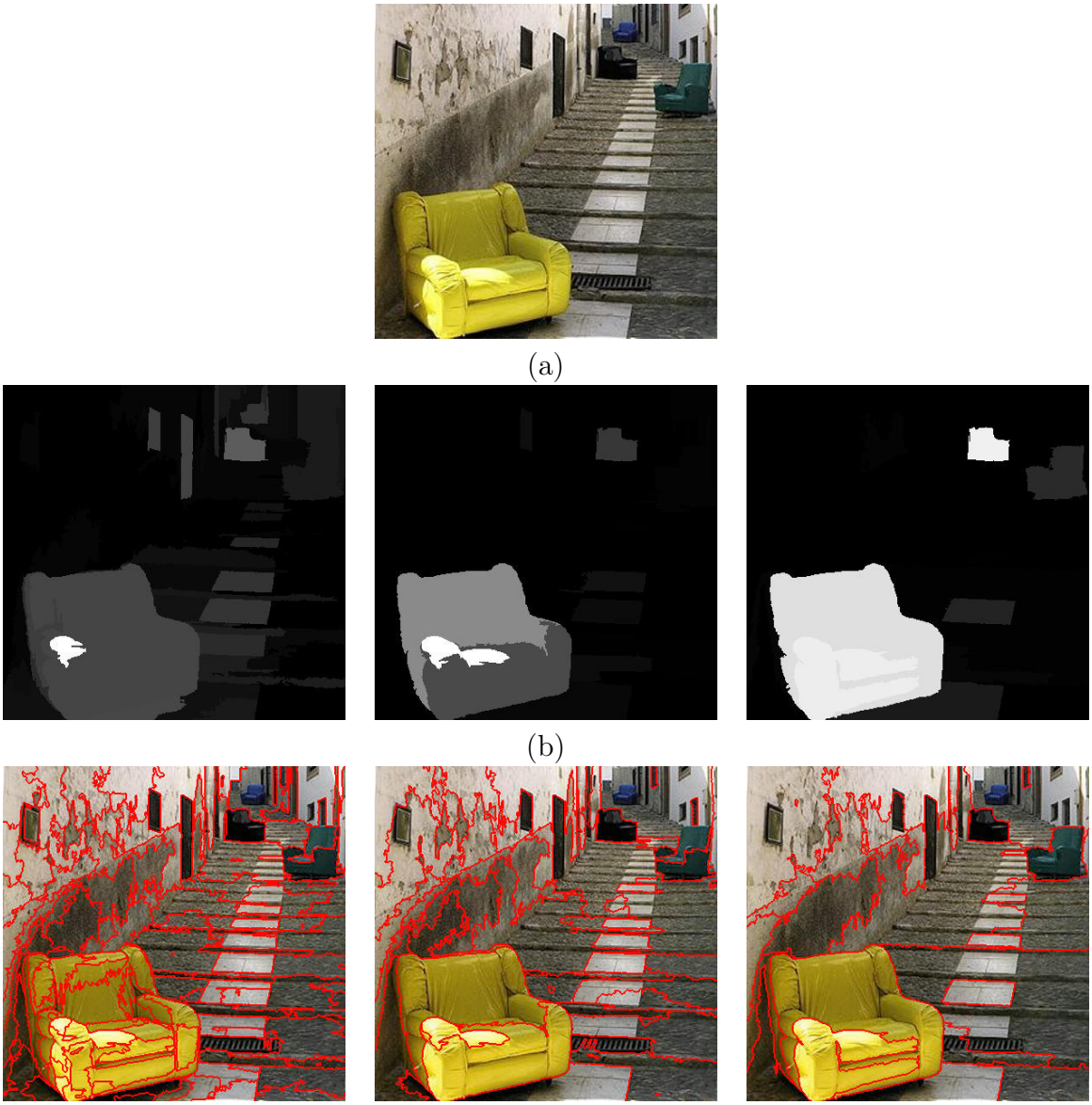


Figure 1.6: (a) Original image. (b-c) From the left to the right, results and superpixel segmentation of ITSELF's iterations 1, 5 and 8, respectively. Note how the number of superpixels change to incorporate multiple scales.

are presented on Chapter 4, with example strategies of queries selection (Section 4.5) and priors estimation (Section 4.4). Some of ITSELF's advantages and short-comes are presented in Section 5, together with comparative results between ITSELF and other saliency estimators. Lastly, conclusions are drawn on Section 6.

Chapter 2

Theoretical Background

In this chapter we are going to start with the notations and definitions (Section 2.1) used within this work. Then, we introduce the *Image Foresting Transform* in Section 2.2, which is a graph-based framework used in this work to estimate object-based superpixels. The method for estimating the superpixels is described in Section 2.3 with it's object based extension being described on Section 2.5. Additionally, we summarize how superpixels can be used to compute saliency from region contrast in Section 2.4. Finally, we present a method to integrate multiple saliency maps using a cellular-automata framework, in Section 2.6.

2.1 Notations and Definitions

We define saliency as the quality that allows an object to stand out from its surroundings. Then, one can specify how salient an object is by attributing them a saliency score. A saliency estimation algorithm attributes to each pixel in an image a saliency score. We store these pixel saliency values in a saliency map, which is here described as a pair $SM = (P, \boldsymbol{S})$, in which $P \subseteq \mathbb{Z}^2$ is the set of pixels, and $\boldsymbol{S} : P \to \mathbb{R}^1$ maps a saliency score to each pixel.

Similarly, we represent any image as a pair $I = (P, \boldsymbol{I})$, in which $\boldsymbol{I} : P \to \mathbb{R}^m$ define the values of the image channels. In this work, we consider colored and grayscale images — *i.e.* $m \geq 1$. Similarly, we represent a saliency map as a pair $SM = (P, \boldsymbol{S})$, in which $\boldsymbol{S} : P \to \mathbb{R}^1$ maps a saliency score to a pixel. Also, let a radius r define an adjacency size, and $\mathcal{A} \subset P^2$ denote an adjacency relation of pixels, where $(p, q) \in \mathcal{A} \leftrightarrow \|p - q\| \leq r$. For simplicity, we denote an adjacency with radius $r = \sqrt{2}$ as $\mathcal{A}_4$, with 4 representing the number of adjacent pixels in said relation.

A superpixel segmentation algorithm divide an image into n superpixels — *i.e.* regions of connected pixels that share similar image properties. Let $S \subset P$ be a superpixel and $\mathcal{S} \supset S, \|\mathcal{S}\| = n$ be the super-set of all superpixels. Some superpixels are used during the saliency computation as foreground or background examples and are compared to the other superpixels when estimating the saliency scores. These superpixels are named *query superpixels* and compose the query set $Q \in \mathcal{S}$.

An image I can be also be depicted as either a graph of pixel or superpixels. We

denote a graph of pixels as $\mathcal{G} = (P, \mathcal{A})$, where the pixels are the vertices and the edges are defined by an adjacency relation commonly of size four or eight. The graph of superpixels, on the other hand, is denoted as $\mathcal{G} = (\mathcal{S}, E)$, where the vertices are superpixels and $E = E_{\mathcal{A}} \cup E_T \cup E_Q$, where $E_{\mathcal{A}}$ is the set of *adjacency edges*, E_T is the set of *transitively extended edges*, and E_Q of *query edges*. The query edges connect every superpixel to every query, *i.e.* $E_Q = \{(S, R) \in \mathcal{S} \times Q \mid S \neq R\}$. The adjacency edge set connect every vertex to its adjacents in the image domain, *i.e.* $E_{\mathcal{A}} = \{(S, R) \in \mathcal{S}^2 \mid \exists\ (p, q) \in \mathcal{A}_8$ for $p \in S, q \in R$, and $S \neq R\}$. Lastly, the transitively edges extend the image adjacency by one level, $E_T = \{(S, R) \in \mathcal{S}^2 \mid \exists\ W \in \mathcal{S}$ that $(S, W), (W, R) \in E_{\mathcal{A}}\}$.

A path π is a sequence of distinct vertices $\pi = \langle p_1, p_2, ..., p_k \rangle$, where $(p_i, p_{i+1} \in \mathcal{A}, \forall i \in [1..k)$. A path is *trivial* if it contains only one vertex $\pi = \langle p \rangle$, and non-trivial otherwise. A path with terminus p is denoted as π_p and a path-extension from it's former terminus p to a new terminus q is denoted as $\pi_p \cdot \langle p, q \rangle$.

A path-cost function $\boldsymbol{f}(\pi)$ attributes a value to each path according to image properties of the pixels that compose it. Common defining properties are local image features such as color and texture. The path-cost function represents the connectivity strength between the path's start and terminus, therefore, every connectivity function requires an initialization and a path-extension rule. Two common examples are the additive connectivity function ($\boldsymbol{f}_{sum}$), and the maximum weight function ($\boldsymbol{f}_{max}$), defined as:

$$\begin{aligned} \boldsymbol{f}_{sum}(\langle q \rangle) &= \boldsymbol{h}(q) && (2.1) \\ \boldsymbol{f}_{sum}(\pi_p \cdot \langle p, q \rangle) &= \boldsymbol{f}_{sum}(\pi_p) + \boldsymbol{w}(p, q) && (2.2) \end{aligned}$$

$$\begin{aligned} \boldsymbol{f}_{max}(\langle q \rangle) &= \boldsymbol{h}(q) && (2.3) \\ \boldsymbol{f}_{max}(\pi_p \cdot \langle p, q \rangle) &= max\{\boldsymbol{f}_{max}(\pi_p), \boldsymbol{w}(p, q)\} && (2.4) \end{aligned}$$

where $\boldsymbol{h}$(q) determines a initial value for trivial paths and $\boldsymbol{w}(p, q)$ is the fixed non-negative weight of the edge (p, q).

2.2 Image Foresting Transform

The *Image Foresting Transform* (IFT) is a framework used to implement image processing operators based on optimum connectivity [18]. For a given image graph, a connectivity function $\boldsymbol{f}(\pi_q)$ must be defined for every path $\pi_q \in \Pi_q$ out of all possible paths with terminus q, including trivial paths. The general IFT algorithm fundamentally minimizes a cost map according to the path-costs $\boldsymbol{C}(q) = min_{\forall \pi_q \in \Pi_q}\{\boldsymbol{f}(\pi)q)$, partitioning the graph into an *Optimum-Path Forest* (OPF). Let r the root of a tree, and $\pi_q = \langle r, ..., p, q \rangle$, the forest is represented by a predecessor map, where $\boldsymbol{Pr}(q) = p \in \mathcal{A}(q)$ for every non-root vertices, and $\boldsymbol{Pr}(r) = nil \notin P$ for the roots. Therefore, each optimum path π_q is stored backwards in $\boldsymbol{Pr}$.

Typical IFT applications restrict the optimum-path search to be performed on a set

of starting vertices $Sd \subseteq P$, where its elements are named *seeds* for they start each tree of the forest. In this scenario, the initial connectivity value is defined as $\boldsymbol{h}(q) = 0, \forall q \in Sd$ and $\boldsymbol{h}(p) = +\infty, \forall p \in P \setminus Sd$. If these seeds are labeled and the label is propagated to the other vertices on it's optimum path, the IFT algorithm can perform object and superpixel segmentation.

The IFT algorithm is a generalization of the Dijkstra algorithm using a less restrictive connectivity function [14]. It starts with a predecessor map containing only trivial paths and iteratively extends the paths until every pixel belongs to one tree. Figure 2.1 shows an example of the execution of the IFT algorithm using a connectivity function that assumes the maximum weight along the path (Equation 2.3). During each iteration, a optimum path π'_p is extended by an edge $\pi'_q = \pi_p \cdot \langle p, q \rangle$ if $\boldsymbol{f}(\pi'_p \cdot \langle p, q \rangle) < \boldsymbol{f}(\pi_q)$. The trees grow out of the seeds, aggregating to it the pixels with higher connectivity strength to its root. If a pixel is misplaced in the wrong tree on starting iterations, the better fitting tree will conquer him later on (this event is highlighted by pink circles on Figure 2.1). The general IFT algorithm is described on Algorithm 1.

Algorithm 1 General IFT

Input: Input image $I = (P, \boldsymbol{I})$, an adjacency relation $\mathcal{A}$, and a connectivity function $\boldsymbol{f}$
Output: Predecessor map $\boldsymbol{Pr}$ and its path-cost map $\boldsymbol{C}$

```
for each p ∈ P do
    C(p) ← f(⟨p⟩), Pr(p) ← nil, insert p in Q.
end for
while Q ≠ ∅ do
    Remove p from Q, where p = argmin_{∀q∈Q}{C(q)}.
    for each q ∈ A(p) ‖ q ∈ Q do
        tmp ← f(π_p · ⟨p, q⟩).
        if tmp < C(q) then
            C(q) ← tmp, Pr(q) ← p
        end if
    end for
end while
return Pr and C
```

As an example of IFT segmentation, Figures 2.2 and 2.3 present differences over the same image being segmented using two path-cost functions ($\boldsymbol{f}_{max}, \boldsymbol{f}_{sum}$). Note that on Figure 2.2, the object has sharp edges in its interior as well as to the background; the inner edges causes the $\boldsymbol{f}_{sum}$ function to considerably sub-segment the object, while the edges between object and background helps preventing leaks on the $\boldsymbol{f}_{max}$ function. On Figure 2.3, however, there are fewer sharp edges inside the object, which causes the $\boldsymbol{f}_{sum}$ to perform considerably better; while the lack of well defined edges (red arrows) between object and background caused the background seeds to conquer part of the object on $\boldsymbol{f}_{max}$.

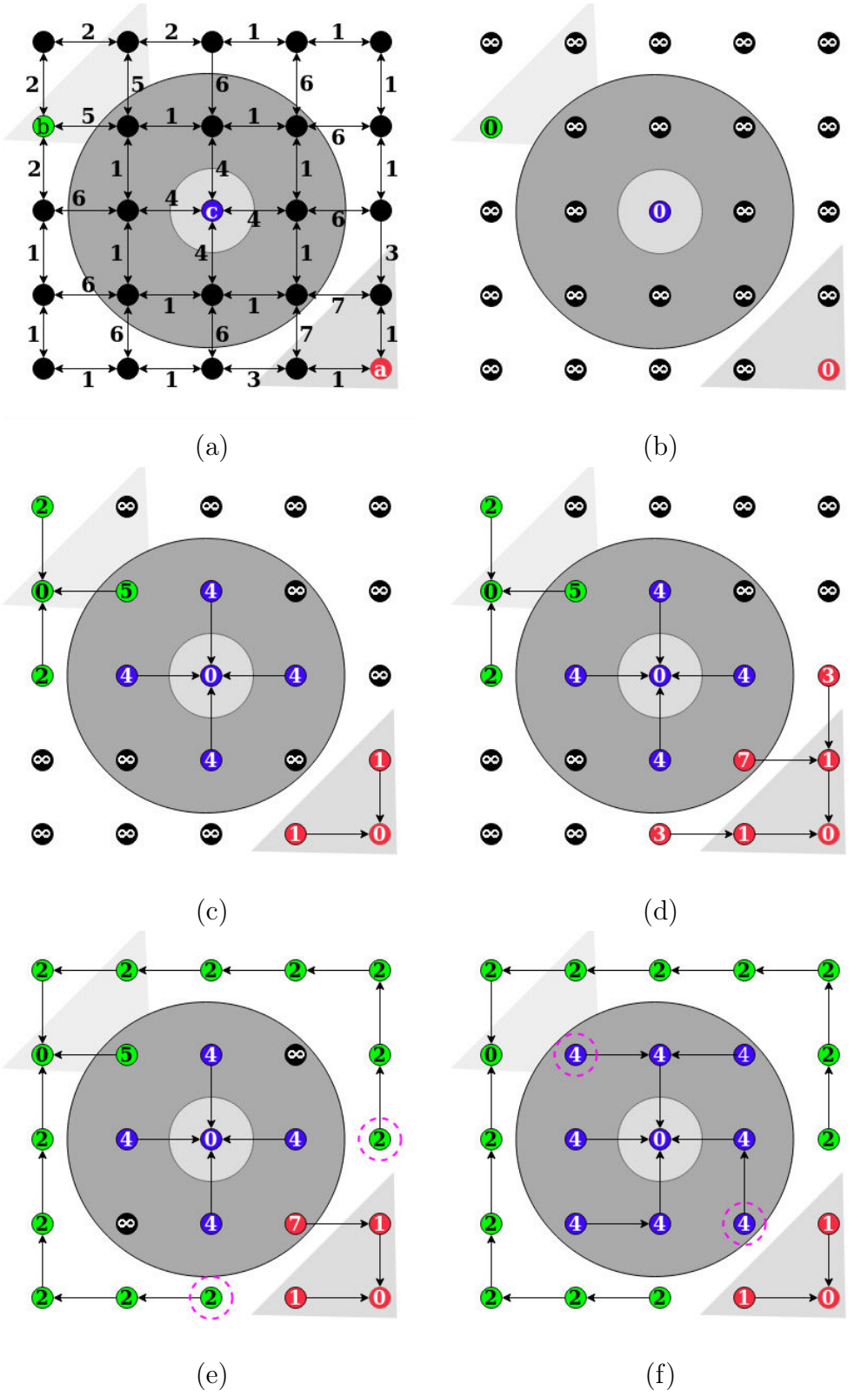


Figure 2.1: (a) The image graph with three starting seeds (a,b, and c); (b) All trivial path are attributed the initialization cost; (c-f) the iterative execution of the IFT algorithm where the seeds conquer the vertices more strongly connected to them.

2.3 Iterative Spanning Forest

By using the IFT framework over uniquely-labeled seeds, Vargas-Muñoz *et al.* [48] proposed the *Iterative Spanning Forest* (ISF) framework for segmenting images into superpix-

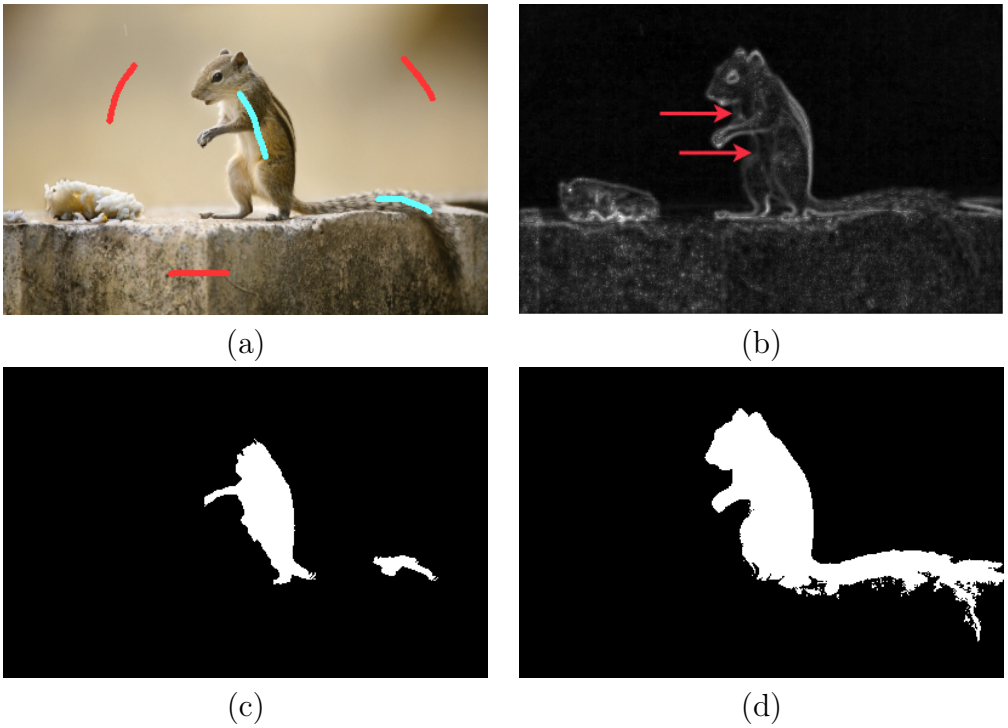


Figure 2.2: (a) Input image with object seeds on blue and background seeds on red. (b) The gradient map used to compute the function weights; (c-d) The segmentation result using the IFT over the $\boldsymbol{f}_{max}$ and $\boldsymbol{f}_{sum}$ functions, respectively. The red arrows point to low gradient areas between foreground and background

els. The framework computes a spanning forest by iteratively improving delineation over enhanced sets of seeds, taking each resulting tree as a superpixel. For such, the framework is composed of three steps: (i) estimate a representative pixel for each superpixel (seeds); (ii) run the IFT over the seed set to create each tree; (iii) recompute the seeds. Through n iterations, the segmentation result is iteratively improved through subsequent executions of steps (ii) and (iii).

To estimate good superpixel seeds, the authors propose two *seed sampling strategies*: *grid*-based estimation, that focus on spreading the seeds evenly, which helps creating more regular superpixels; and a *mix* between grid and entropy-based sampling, which often reduces the over-segmentation of homogeneous image regions. For the *grid*-based sample, an optimum distance between the seeds is estimated based on the number of superpixels desired and the seeds are spread as equally as possible throughout the entire image. The *mix* strategy tries to further divide more heterogeneous regions, better capturing objects characteristics. For such, the authors propose using a two-level quad-tree representation of the image and compute a *Normalized Shannon Entropy* ($\boldsymbol{NSE}(Q)$) value for each quadrant Q. The entropy function is defined as follows:

$$\boldsymbol{NSE}(Q) = \frac{\sum_{i=1}^{n_i} \rho_i log_2(\rho_i)}{log_2 n_i} \tag{2.5}$$

The NSE is initially computed for all first-level quadrants of the image, together with its the mean $\mu(\boldsymbol{NSE})$ and standard deviation $\sigma(\boldsymbol{NSE})$. Whenever the entropy of a quadrant is greater than the mean by one standard deviation — *i.e.* $|\boldsymbol{NSE}(Q) -$

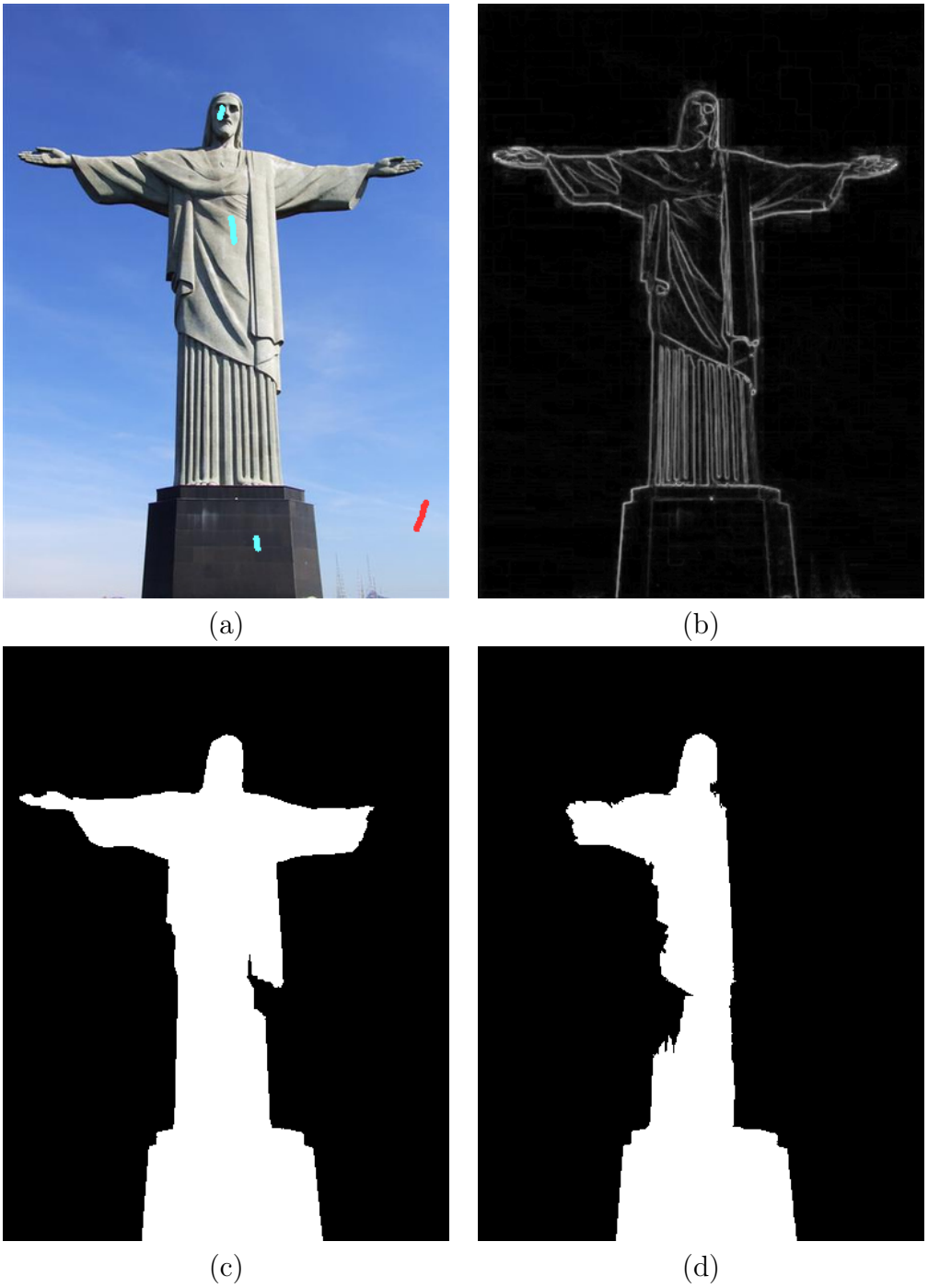


Figure 2.3: (a) Input image with object seeds on blue and background seeds on red. (b) The gradient map used to compute the function weights; (c-d) The segmentation result using the IFT over the $\boldsymbol{f}_{max}$ and $\boldsymbol{f}_{sum}$ functions, respectively

$\mu(\boldsymbol{NSE})| > \sigma(\boldsymbol{NSE})$ — the quadrants are divided into the next level. Afterwards, a entropy score is assigned to the second-level quadrants and the number of seeds inside each quadrant is estimated according to its $\boldsymbol{NSE}$. Finally, the seeds are selected using the grid sampling strategy inside each quadrant.

For delineating the superpixels, different adjacency relations and connectivity functions can be used to change the IFT results. Regarding adjacency relations, the more

common ones are 4- or 8- neighborhood ($\mathcal{A}_4 and \mathcal{A}_8$). The authors propose three connectivity functions based on the $\boldsymbol{f}_{sum}$ ($\boldsymbol{f}_1$, $\boldsymbol{f}_2$) and $\boldsymbol{f}_{max}$ ($\boldsymbol{f}_3$). All of them use the same trivial-path initialization rule given by.

$$\boldsymbol{f}_*(\langle p \rangle) = \begin{cases} 0 & \text{if } p \in Sd \\ +\infty & \text{otherwise} \end{cases} \tag{2.6}$$

Let $S \subset P$ be a superpixel, and $s \in S$ be the root (seed) of S. The difference amoung the functions are in the path-extension cost, where each function is defined as follows:

$$\boldsymbol{f}_1(\pi_p \cdot \langle p, q \rangle) = \boldsymbol{f}_1(\pi_p) + (\|\boldsymbol{I}(q), \boldsymbol{I}(p)\|\alpha)^\beta + \|p, q\|, \tag{2.7}$$

where $\alpha \geq 0$ and $\beta \geq 0$ are parameters to control the compromise between superpixel regularity and border adherence.

$$\boldsymbol{f}_2(\pi_p \cdot \langle p, q \rangle) = \boldsymbol{f}_2(\pi_p) + (\|\boldsymbol{I}(q), \boldsymbol{M}(S)\|\alpha)^\beta + \|p, q\|, \tag{2.8}$$

where s $\boldsymbol{M}(S)$ is the mean color inside the superpixel in a previous iteration (with $\boldsymbol{M}(S) = \boldsymbol{I}(\mathrm{s})$ at the first iteration).

$$\boldsymbol{f}_3(\pi_p \cdot \langle p, q \rangle) = max\{\boldsymbol{f}_3(\pi_p), \nabla(q)\}, \tag{2.9}$$

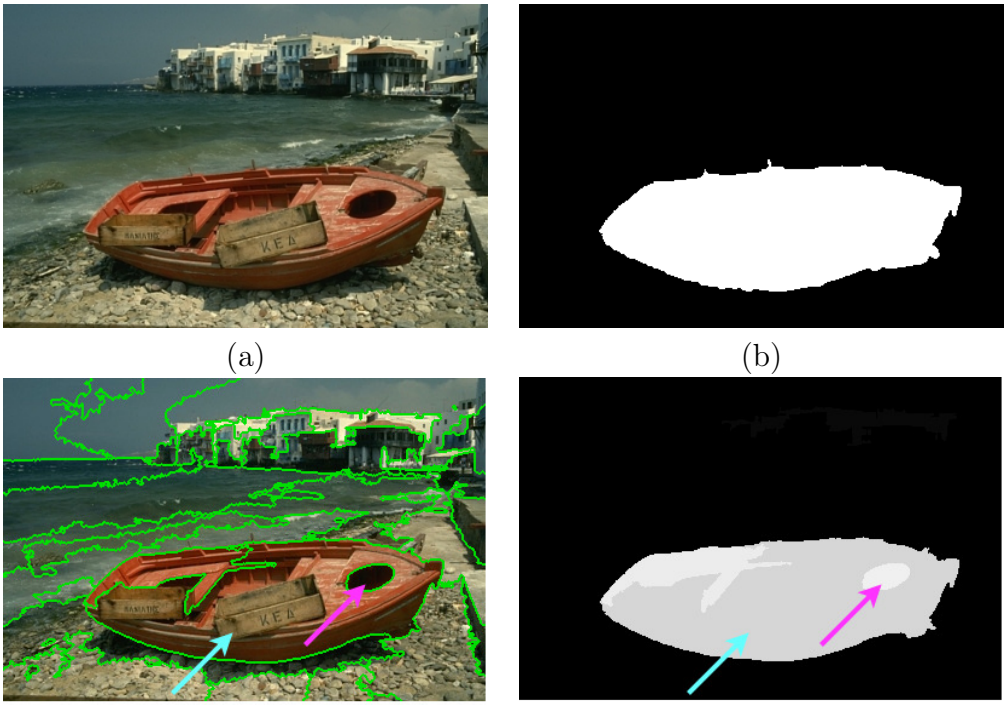
where $\nabla(q)$ is the image gradient in pixel q. As described by the authors, using the maximum weight in the path usually grants high adherence to the image boundaries, and often create less regular superpixels. The first two functions allow for a user-defined compromise between boundary adherence and superpixel regularity thanks to the spatial connectivity component ($\|p, q\|$) controlled by the α and β parameters.

Lastly, the authors propose an automated seed recomputation strategy. Let $s^t \in S$ be the superpixel seed at iteration t. The improved seed is taken as the pixel with most similar color to the superpixel mean color (*i.e.*, $s^{t+1} = \mathrm{argmin}_{\forall p \in S}\{\boldsymbol{I}(p) - \boldsymbol{M}(S)\}$ or the pixel closest to the superpixel geometric center. With the improved seed set, the IFT algorithm is run again and new superpixels are delineated.

2.4 Superpixel-based Saliency Estimation

On an early work, Cheng *et al.* [13] propose using global color contrast to define saliency. The idea is that, image regions that have high color contrast to the others should be considered more salient. For such, they present a histogram-based method (HC) and improve it by adding spatial information via superpixel segmentation (RC). Let $I = (P, \boldsymbol{I})$ be an image, $p \in P$, and C be the set of all unique colors that compose I. Pixel contrast is defined as its dissimilarity to all other pixels of the image $\boldsymbol{S}(p) = \sum_{\forall q \in P} \|\boldsymbol{I}(p), \boldsymbol{I}(q)\|$. To reduce the computational effort, contrast can be defined directly in terms of color $\boldsymbol{S}(p) = \boldsymbol{S}(c_i) = \sum_{\forall c_j \in C} \rho(c_j)\|c_i, c_j\|$, where $\rho(c_j)$ is the percentage of pixels of color c_j in the image, and $c_i = \boldsymbol{I}(p)$.

The color contrast concept is then extended to region contrast by using superpixels: A superpixel containing more contrasting colors should be considered more salient. As



(a) (b)

(c) (d)

Figure 2.4: (a,b) Input image and ground-truth. (c) Superpixel segmentation with thirty two superpixels; (d) a possible resulting saliency map

an example, in Figure 2.4, we segmented the image into a few superpixels to improve clarity. Note how the object is mainly composed of contrasting colors (red and black) to the background (gray, light-brown, blue and white); however, inside the object, there are less contrasting colors (cyan arrow) and more contrasting ones (pink arrow), which impacts on the saliency score of the superpixels containing them.

Let $c_S \in C_S$ denote a unique color inside the region $S \in \mathcal{S}$, and $ce_S \in P$ be the centroid of the superpixel S. The saliency of a region is defined based on its color contrast to all other regions in the image:

$$\boldsymbol{S}(S) = \sum_{\forall R \in \mathcal{S}} \exp^{\frac{\|ce_S, ce_R\|}{\sigma^2}} |R| \boldsymbol{C}(S, R), \tag{2.10}$$

where $|R|$ is used to increase contrast to larger regions, the $\exp^{\frac{\|ce_S, ce_R\|}{\sigma^2}}$ term increases the importance of closer regions, and $\boldsymbol{C}(S, R)$ is the contrast between every combination of colors contained in two regions, defined as:

$$\boldsymbol{C}(S, R) = \sum_{\forall c_S} \sum_{\forall c_R} \rho(c_S, S)\rho(c_R, R)\|c_S, c_R\|. \tag{2.11}$$

Although the spatial distance weight reduces the degrading effect of comparing every superpixel to all others, the computational effort is onerous and unnecessary. To avoid this issue and better represent the relationship among the superpixels, several methods have opted to use superpixel-graphs to represent the image (further discussed in Section 3.3). Superpixel graphs allow for contrast only to be computed between two nodes connected by

an edge, reducing the number of required computations significantly compared to global methods.

In addition to adjacency-based contrast, graph-based saliency methods also utilize *query superpixels* to define contrast. Query superpixels are foreground or background examples, so regions similar to the potential foreground and dissimilar to the background should have a higher saliency score. The queries can be estimated using prior domain knowledge (*e.g.,* one may assume most of the image's borders can be used as background examples), user-placed scribbles, or even another saliency method.

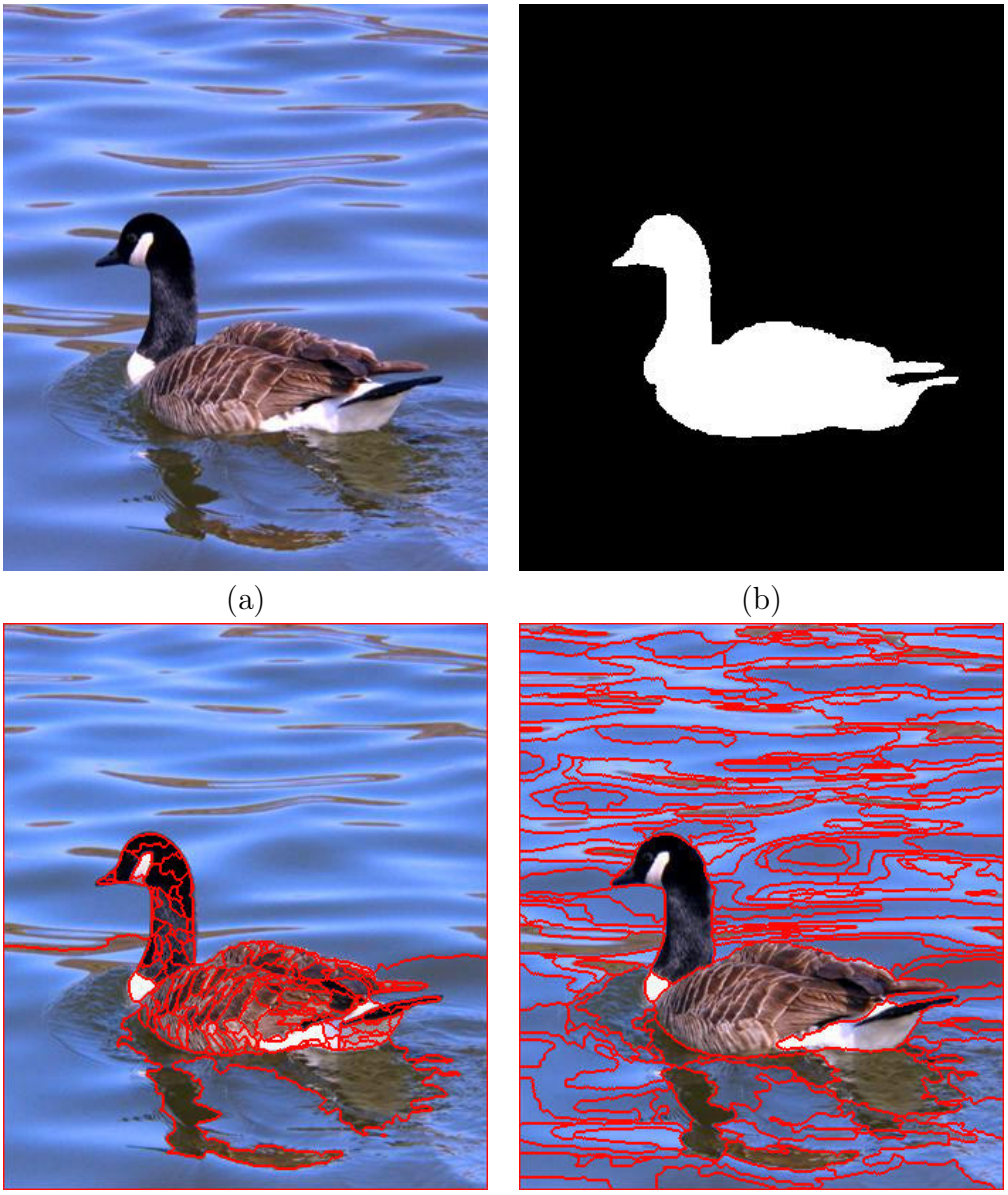
2.5 Object-based Iterative Spanning Forest

Belem *et al.* [5] extended the *Iterative Spanning Forest* (ISF) framework by incorporating object information to improve object representation and delineation. Like its precursor, the *Object-based Iterative Spanning Forest* (OISF) runs multiple iterations of the IFT algorithm over an improved seed set. However, they add object information (often represented by saliency maps) on all three steps (seed estimation, object delineation, and seed recomputation).

So far, two object-based seed sampling strategies have been proposed: the Object-based grid (OGRID); and the Object Saliency Map sampling by Ordered Extraction (OSMOX). Both strategies allow for user control over the number of seeds inside the foreground and inside the background (Figure 2.5).

The OGRID strategy consists of placing evenly spaced seeds inside each component of the binary image. The number of seeds inside a component depends on its area, with larger components receiving a larger number of seeds. Let n be the number of superpixels, $\rho(c) \in (0, 1)$ be the number of seed inside each component, $SM = (P, \boldsymbol{S})$ be a saliency map, δ be a threshold, and $\hat{I} = (P, \boldsymbol{B})$ be a binary image where $\boldsymbol{B}(p) = 1$ if $\boldsymbol{S}(p) > \delta$ and $\boldsymbol{B}(p) = 0$ otherwise. Each component $C_i \subset P, i = 1, ...c$ on $\hat{I}$ receives $n_i = \frac{\rho(c)n|C_i|}{\sum_{i=1}^{c}|C_i|}$ seeds equally spaced using a *geodesic seed sampling strategy.*

Note that on OGRID, the object information is flattened by the threshold and is only used to determine the number of seeds inside each component. To account for the information loss, the authors proposed OSMOX [6]. The strategy consists of selecting seeds from a priority queue of pixels, where the priority is defined according to the saliency of the pixel's neighbors. To ensure better seed distribution, each pixel selected as a seed, the priority of its adjacent pixels is reduced, and the priority queue is rearranged. The previous steps are repeated until the number of seeds is obtained. For such, let $\rho(o) \in (0, 1)$ be the percentage of object seeds, $n_o = \rho(o)n$ be the number of object seeds, $d = \sqrt{\frac{|P|}{n_o}}$ determine the adjacency radius for priority computation, and $\mathcal{A}_d$ be the adjacency relation obtained using d. Each pixel is placed into a priority queue Q, ordered by their priority $\boldsymbol{PR}(p) = \sum_{\forall q \in \mathcal{A}_d(p)} \boldsymbol{S}(q)$. The pixel s with the highest priority is removed from Q and inserted on the seed set S. Then, the priority of every pixel $a \in \mathcal{A}_d(s)$ is reduced, resulting on $\boldsymbol{PR}(a) \leftarrow (1 - \exp^{\frac{-\|s,a\|^2}{2\sigma^2}})\boldsymbol{PR}(a)$. This process is repeated until the desired number of seeds is reached or $Q = \emptyset$. The same algorithm is run on the complement of the saliency map to obtain the background seeds.



(a) (b)

(c) (d)

Figure 2.5: (a) Input image. (b) Ground-truth; (c) Super-segmented salient object [5]; (d) Sub-segmented salient object. Both segmentation images were computed using OISF [5] with 200 superpixels and the ground-truth as the input saliency map

For their connectivity function, the authors propose an extension of Equation 2.7, but taking object information into consideration:

$$\boldsymbol{f}_o(\pi_p \cdot \langle p, q \rangle) = \boldsymbol{f}_o(\pi_p) + \|p, q\| + \left[\alpha \|\boldsymbol{I}(q), \boldsymbol{I}(p)\| \gamma^{|\boldsymbol{S}(r_p) - \boldsymbol{S}(q)|} + \gamma^{|\boldsymbol{S}(r_p) - \boldsymbol{S}(q)|}\right]^{\beta}, \quad (2.12)$$

where $\gamma > 0$ controls the influence of the saliency map in the boundary adherence. The addition of object information in the path-cost function improves the delineation, especially on foreground-to-background transitions with a low gradient. Figure 2.6 exemplifies how the IFT is used to delineate superpixels and how object information can be used to represent the objects better. Using only color information, the red-labeled

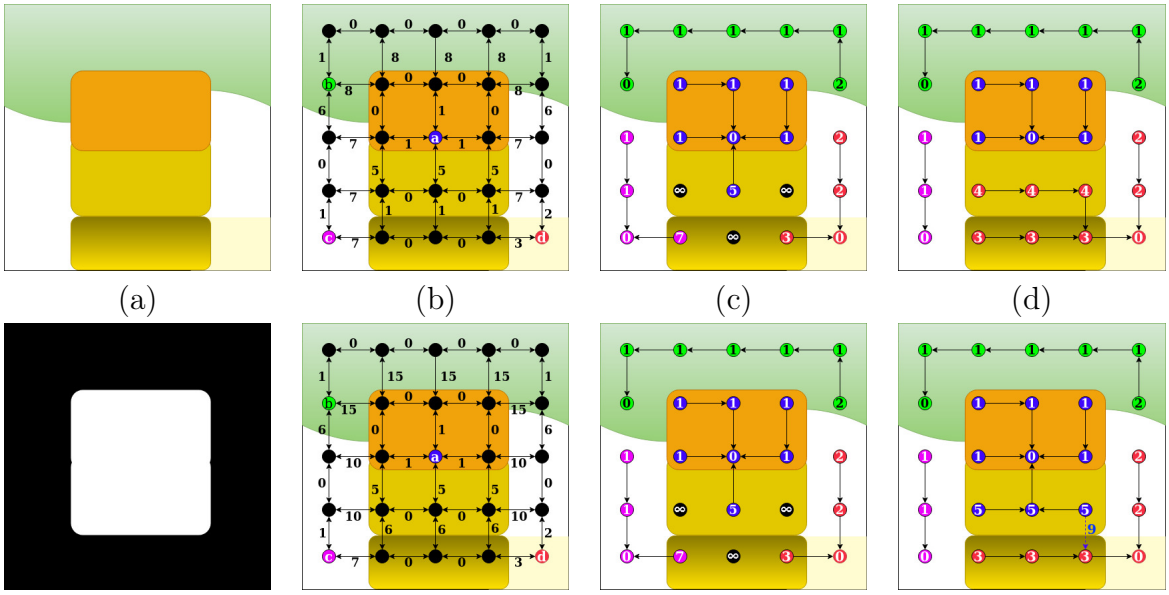


Figure 2.6: (a,e) The original image and its saliency map, respectively; (b,f) The cost maps using only color difference and combining color difference to object information, respectively; (e-f) The result of initial iterations that was not effected by the object information. (g) The resulting superpixels using only color difference; (h) The resulting superpixels combining color difference and object information.

background superpixel conquered the similar adjacent part of the object. However, by adding the saliency map information, the path-cost weight to cross the object boundary was increased, resulting in the correct object delineation. Figure 2.7 shows how different settings of α, β, and γ change the behavior of the algorithm.

For the seed recomputation strategy, they use the same strategy as regular ISF, but extend the color vector in one dimension by adding the pixel's saliency value.

2.6 Cellular Automata for Saliency Map Integration

A cellular automaton is a model of a grid system of cells, where the cells change its state over time (evolve) according to an update rule that consider the state of the cell's adjacency. We denoted an automaton as a grid $G = (\mathcal{C}, \boldsymbol{S}^t_{\oplus})$, where $\mathcal{C}$ is the set of all cells and $\boldsymbol{S}^t_{\oplus} : \mathcal{C} \rightarrow \mathbb{R}^1$ maps a value to each cell in time t.

Let $\mathtt{c} \in \mathcal{C}$ be a cell and $\mathcal{A}_4(\mathtt{c})$ be the set of cells inside the 4-adjacency of $\mathtt{c}$. To illustrate in a simple example, say an automaton is used to represent a two-dimension binary image, where each cell $\mathtt{c} \subseteq \mathbb{Z}^2$ represents a pixel with only two possible states ($\boldsymbol{S}^t_{\oplus}(\mathtt{c}) = \{0, 1\}$)). The initial state of a cell ($\boldsymbol{S}^0_{\oplus}$) is determined by the value of the pixel that it represents and the subsequent states depend on a update function and the state of the cells adjacency. For such, let's define an update function to be:

$$\boldsymbol{S}^{t+1}_{\oplus}(\mathtt{c}) = \begin{cases} 1 & \text{if } \sum\limits_{\forall \mathtt{a} \in \mathcal{A}_4(\mathtt{c})} \boldsymbol{S}^t_{\oplus}(\mathtt{a}) \geq 2 \\ 0 & \text{otherwise.} \end{cases} \tag{2.13}$$

Figure 2.7: (a,b) The original image and its saliency map, respectively; (c,d) OISF segmentation insuring better superpixel regularity ($\alpha = 12$, $\beta = 0.8$) and better boundary adherence ($\alpha = 0.8$, $\beta = 12$; (e,f) OISF segmentation using different object information weight on delineation ($\gamma = 0.5$ and $\gamma = 2.0$). The cyan arrows indicate regions where the object information allowed separation of low-contrast boundaries between foreground and background.

If this function is used as the update rule of a binary image, some hollow objects that are fully enclosed will start getting filled over each of the automaton's iterations, as shown in Figure 2.8.

Similar to graphs or matrices, cellular automata are generic models that can be used for multiple purposes. Yao Qin *et.al.* [41] proposed an iterative saliency-estimation algorithm that uses Cellular-Automata and a Bayesian framework to combine multiple saliency maps. In this work, we are interested on the integration framework. In their proposed integration automaton, the cells are the pixels of all saliency maps stacked on the z-axis,

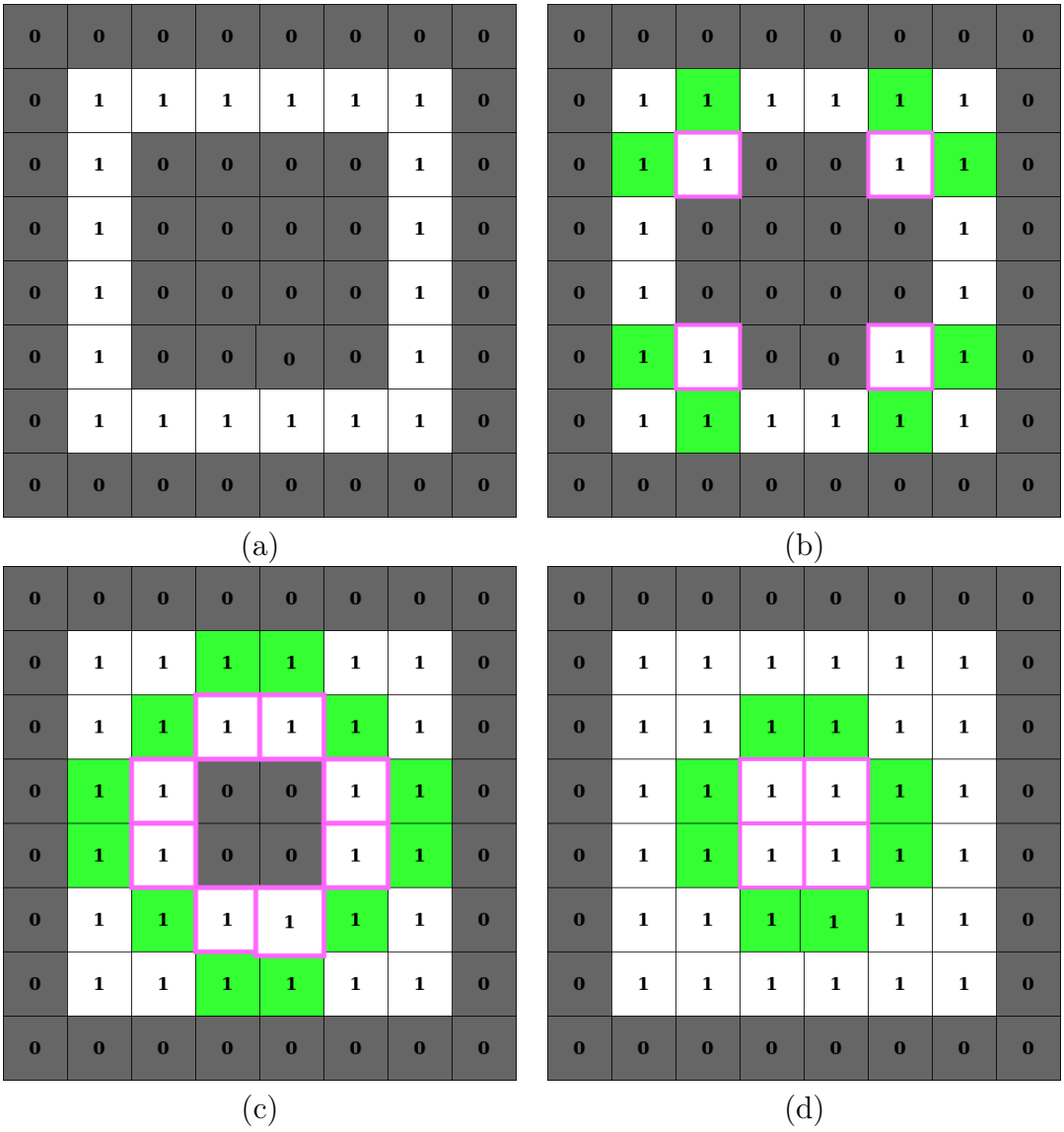


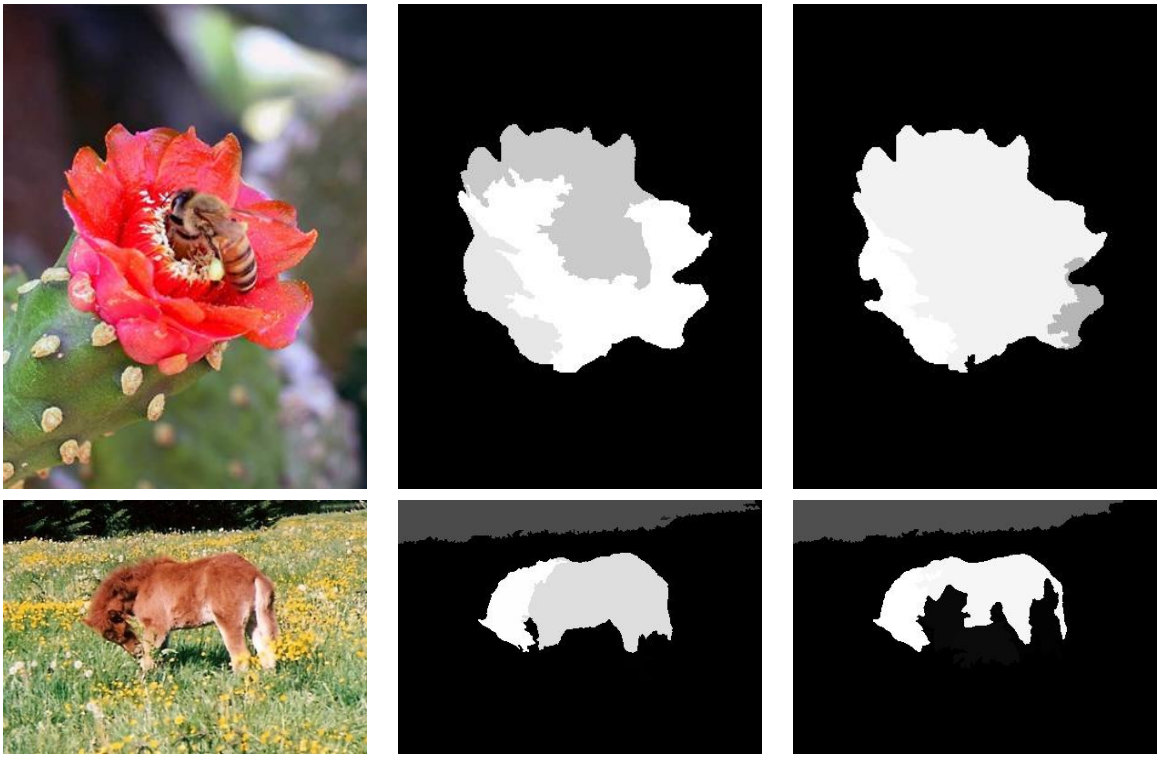
Figure 2.8: (a) The original automaton cells derived from a fictional image, where a hollow white square contains an inner background square (in gray); (b-d) The inner square start changing state (depicted by the pink borders) due to the state of it's adjacent object cells (in green).

forming a 3D grid $G = (\mathcal{C}, \boldsymbol{S}^t_{\oplus})$. Formally, an ordered list of saliency maps $\langle SM_1..SM_m \rangle$, $\mathcal{C} = \{(x_p, y_p, i) \in P^2 \times \mathbb{N}^* \mid p = (x_p, y_p) \in P_i\}$ and $\boldsymbol{S}^1_{\oplus}(\mathsf{c}_i) = \log(\boldsymbol{S}_i(p))$, for $SM_i = (P_i, \boldsymbol{S}_i)$, and $1 \leq i \leq m$.

Let $\mathsf{c}_i = (x_p, y_p, i)$, $\mathsf{c}_j = (x_q, y_q, j) \in \mathcal{C}$. The automaton evolution is defined over a *cuboid adjacency relation* $\mathcal{A}_{\square}$. We formally define the cuboid adjacency to be $\mathcal{A}_{\square} = \{(\mathsf{c}_i, \mathsf{c}_j) \in \mathcal{C} \mid \exists\ (p, q) \in \mathcal{A}_4$ which $p = (x_p, y_p)$ and $q = (x_q, y_q)\}$.

A cell c_i is considered salient if its saliency score is higher than the mean saliency value (μ_i) of its originating map SM_i. The update rule defines that a cell will have its saliency changed according to how consistently salient its adjacency is:

$$\boldsymbol{S}^t_{\oplus}(\mathsf{c}_i) = \boldsymbol{S}^{t-1}_{\oplus}(\mathsf{c}_i) + \Lambda \sum_{\forall \mathsf{c}_j \mid (\mathsf{c}_i, \mathsf{c}_j) \in \mathcal{A}_{\square}} sign(\boldsymbol{S}^{t-1}_{\oplus}(\mathsf{c}_j) - \mu_j) \tag{2.14}$$



(a) (b) (c)

Figure 2.9: (a) Input image. (b-c) Result of saliency map integration using $\lambda \in \{0.01, 0.1\}$, respectively. Note that although a higher λ creates more homogeneous salient objects, less salient object parts may be lost.

where $\Lambda \in (0..1]$ is a constant that controls the strength of the update (Figure 2.9) and the summation defines a score for how consistently salient the cell's adjacency is. The final updated saliency score of each pixel on each map is than normalized: $\boldsymbol{S}^t_{\oplus}(\mathsf{c}_i) \leftarrow \frac{\exp^{\boldsymbol{S}^t_{\oplus}(\mathsf{c}_i)}}{(1+\exp^{\boldsymbol{S}^t_{\oplus}(\mathsf{c}_i)})}$.

After t iterations, the final saliency map is the cell combination on the z coordinate:

$$\boldsymbol{PS}(p) = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{S}^t_{\oplus}(\mathsf{c}_i), \tag{2.15}$$

where $\mathsf{c}_i = (x_p, y_p, i)$.

Chapter 3

Related Works

On this chapter, we will present the state-of-the-art on saliency estimation, covering supervised and unsupervised methods. On Section 3.1, we present the current state-of-the-art supervised methods, as well as the strategies used to overcome it's short-comes; we also discuss how the recent saliency estimation literature mostly contains deep-learning based methods and how unsupervised saliency estimation can assist these tasks. On Section 3.2 we present the state-of-the-art methods and common strategies for classic unsupervised saliency estimation. Because we model our saliency model using a superpixel-graph, Section 3.3 present multiple graph-based unsupervised saliency estimation strategies. Section 3.4 addresses superpixel segmentation using the Image Foresting-Transform framework, which implements a method that allows the incorporation of saliency maps to improve superpixel delineation. Section 3.5 present how superpixels have been used in saliency estimation. Lastly, we draw conclusions (Section 3.6) over the current state of the saliency estimation literature.

3.1 Supervised Saliency Estimation

Saliency estimation algorithms can be categorized as either supervised or unsupervised. Supervised estimators learn discriminant salient features from ground-truth images, while unsupervised approaches use heuristics and prior knowledge to create observation-based saliency models. The usage of supervision often result on better saliency maps at the cost of having an extensive number of annotated data, which is easier to come by on natural images but poses a problem on other image domains such as biomedical images.

According to a recently published survey [8], the recent methods are in its majority supervised algorithms, with the state-of-the-art being achieved by methods based on *deep Convolutional-Neural-Networks* (CNN). The classic deep-learning based methods used convolutional layers to extract meaningful salient-related features from small regions and then classify them using a *Multi-Layer Perceptron* (MLP) [23, 49, 62, 28]. However, the MLP-based strategy does not preserve spatial information. After the seminal work of Long *et al.* [34], *Fully Convolutional Neural Networks* started being used for saliency estimation [56, 61, 11, 40], allowing for spatial relation to be maintained during the entire process.

Regarding the classic CNN models, the most common approach is to combine two networks, where the first network focus on high-level features to roughly detect the salient objects and the second tries to find a refined segmentation. These methods often utilize superpixels-based operations to improve the object boundaries and create local context.

The FCN methods also utilize multiple networks, usually having one network to identify global saliency and a second one local saliency to capture smaller object details. Recent multiple-FCN methods allows for deep models to be suitable for computing high resolution saliency maps [56, 61, 11] and considerably improve the boundary between object and background [61, 40].

All the above methods, however, require a large number of human-annotated images to be trained. Their models are pretrained using the ImageNet dataset [16] and fine tunned on the specific datasets, usually requiring around ten thousand images, as reported in [50]. Specially when shifting image domains, not only acquiring a large enough annotated dataset is rarer, but pretraining on the natural-images from ImageNet provides is less impactful result, as shown in [42].

A few methods propose unsupervised deep-learning strategies. However, these methods still require pixel-level annotation, they just propose utilizing saliency estimation instead of human annotation to obtain the masks. The saliency maps can be provided either from a pre-trained CNN [30] or from classic unsupervised saliency estimators [58, 57]. In this regard, creating better unsupervised saliency estimators for non-natural images can greatly benefit such approaches.

The success of deep-learning methods shifted the majority of the focus to supervised methods: Even though there is a vast literature on classic unsupervised saliency estimators, the most recent non-deep-learning unsupervised method reported on the aforementioned saliency survey [8] dates five years back from the survey's published year. The next sections present an overview of unsupervised saliency estimations, focusing on methods more closely related to our proposal.

3.2 Unsupervised Saliency Estimation

Most unsupervised saliency estimators model saliency using a combination of prior domain-specific knowledge, and salient characteristics extracted from the input image. The prior knowledge is used to create global assumptions that do not depend on image-specific characteristics: On natural images, for example, the salient object is most likely centered [39, 44], focused [27] and composed of vivid colors [39, 44]. It is not hard to imagine scenarios where these assumptions fail, and the results are sub-par.

On the other hand, bottom-up information can be used to model saliency based on similarities of intrinsic low-level image information: For example, one may assume that regions with high color contrast to its adjacency are likely to be salient [13]. However, the over-segmentation of the regions (superpixels) may introduce errors. In this regard, several methods propose saliency to be computed on multiple scales [31, 59, 46, 26], and then combined later on.

An example of bottom-up multi-scale method is the *Discriminative Regional Feature*

Integration (DRFI) [26]. On DRFI, a region is deemed salient if it has a high contrast to the rest of the image according to a weighted distance of their feature vectors. The feature vectors are composed of color information — the region's colors in RGB, HSV and L*a*b, as well as the histogram histograms of the color-spaces — and texture information obtained through the responses of an LM filter bank [29]. Even though the features are combined with different weights learned using a Random Forest, the features themselves are not learned through supervision, and require only a few labeled images to train the model, thus making it more comparable to unsupervised methods than supervised ones. Even though the method was proposed in 2013, it is still used on deep-learning benchmarks as one of the state-of-the-art classic saliency estimation methods.

Another common approach is to define global query regions to act as first estimates for background and foreground. Assuming the object to be usually centered and fully enclosed on natural images, the most common query selection strategy is to use regions on the image limits as potential background. To mitigate the error on images where the object does touch the image border, multiple strategies have been proposed to reduce the influence of miss-selected queries. One strategy is to combine multiple saliency maps using subsets of the boundary regions, say one map for each of the four sides (top, right, bottom, and left) [59, 60]. An alternative is to assign a confidence value to boundary-regions based on how much of its adjacency is connected to the image border [63]. Although these strategies reduce the error caused by foreground regions touching the image limits, they still do not perform well on images where the object occupies most of the potential queries. We explore graph-based strategies further on Section 3.3.

Instead of assuming the background to be on the image borders, another set of algorithms expect the image background to be composed of highly redundant information. They solve saliency estimation using *low-rank matrix recovery* (LR) theory. LR-based methods use a low-rank feature matrix to approximate the background regions, and sparse salient object regions, on the other hand, are represented by a sparse sensory matrix. One method that stands out on this approach uses a *Structured Matrix Decomposition (SMD)* [39] model, which adds connectivity constraints and a regularization step used to assist images with a cluttered background. SMD also utilize top-down information by modeling location, color and background priors: A region is considered salient if they are close to the image center, they contain red and yellow tones, and if they are not within the image limits. Using a fully unsupervised model, they reported better results than DRFI.

Note that all approaches have to make assumptions based on prior knowledge of the image domain. In this regard, by pre-selecting a set of query strategies and top-down priors, even bottom-up strategies are constrained to specific scenarios.

3.3 Graph-based saliency estimation

In recent years, many methods have been proposed using graphs to model saliency [63, 46, 59, 54, 52]. Each image is represented by a graph, where the vertices are image regions (superpixels), and an edge connects two related vertices. Regions are usually connected to their adjacency and to query regions, and saliency is estimated in a bottom-up manner.

Yang *et.al.* [54] proposed using the four image borders as background queries to compute four saliency maps using manifold ranking. Even though the multiple maps strategy reduces the error compared to using all background regions simultaneously, the resulting combination commonly highlights only parts of the salient objects. As a further step, the authors threshold the resulting background-map combination to use as a foreground query to estimate the final saliency map. Wu *et al.* [52] use a similar framework, but they further improve the background queries by estimating how salient each border region is amongst themselves.

Zhu *et al.*[63], instead of computing multiple maps, propose a weighting function to determine the confidence of a border region to be part of the background. They also include a smoothness term to regularize the optimization of cluttered regions of the image.

Taking closer attention to the role superpixels have in the process, Tong [46] proposed computing saliency on multiple scales of superpixels and added a filtering property to improve edge preservation on the resulting map. They compute multiple single-scale saliency maps and integrates them by proposing an integration function that optimizes a pixel-to-superpixel similarity measure. Similarly, Zhang *et al.* [59] propose using multiple scales of superpixels to compute a background and foreground-query based hypergraph saliency estimator. They present their results using both foreground and background, or using only one of the two. Combining both queries outperforms both other options. We want to compare our results to the hypergraph estimator, however, we could not reproduce our experiments due to the code not being publicly available. Their reported results, however, does not consistently outperform the methods we are comparing our approach to.

All the aforementioned graph-based strategies use a bottom-up only approach and do not leverage top-down prior knowledge. As shown by Peng *et al.*[39], combining both top-down and bottom-up strategies may be beneficial.

3.4 Superpixel segmentation using the IFT framework

As presented in Section 2.3, the IFT framework can be used to implement superpixel segmentation algorithms when executed on a fit seed set. The first IFT-based superpixel segmentation algorithm, namely IFT-SLIC [3], was an extension of the *Simple Linear Iterative Clustering* (SLIC) [2]. Let n be the number of superpixels desired on an image $I = (P, \boldsymbol{I})$; the IFT-SLIC uses a regular-spaced grid of distance $d_p = |P|/n$ as their seed sampling strategy. For the superpixel delineation, IFT-SLIC uses the IFT framework over the estimated seeds and Equation 2.7 as the path-cost function. With the initial superpixels delineated, the seed set is improved by selecting the pixel whose position is closest to the superpixel center as the new seed.

Later, Vargas-Muñoz *et al.* [48] proposed a generalized framework for superpixel computation using the IFT, namely ISF (Section 2.3). Using ISF, new methods can be created by defining a seed sampling strategy, a path-cost function, and a seed-recomputation strategy. As examples of ISF's flexibility, additional to the object-based version of the framework (Section 2.5, ISF has been used to create methods that provide multiple-

scale superpixel hierarchies (Recursive ISF [20]), that generate symmetrical supervoxels on both brain hemispheres in brain MR images (SymmISF)[36], and that create class-representative superpixels of stacked registered class-specific images [10].

More restrictively, both the SymmISF and the class-specific ISF use object information; however, the OISF provides a more general structure that allows the incorporation of object information via saliency maps, which is a required feature within our proposed framework. There are future explorations to be made regarding possible combinations of OISF's strategies to other ISF-based methods, such as RISF [20], or the newly proposed seed-removing based method entitled *Dynamic and Iterative Spanning Forest* (DISF) [7]. RISF provides superpixel hierarchies that could be used to extract multiple scale information from the image, and DISF has shown improved boundary adherence over most ISF-based segmentations.

3.5 Superpixels for saliency estimation

Early methods used single pixels or nxn-blocks of pixels to compute contrast [47, 24, 1] but they usually lack well-defined separation between object and background, overly increasing the saliency of blocks adjacent to actual salient regions. To better define these regions, most modern methods adopted the usage of superpixels.

Many methods have been proposed to super-segment the image into superpixels (*e.g.* [2, 19, 15, 48]), however, choosing which superpixel segmentation to use is a somewhat overlooked task when estimating saliency. Most methods opt for using SLIC [2], which is a fast grid-based segmentation method that creates regular superpixels (superpixels are of similar size and shape). Despite superpixel regularity being an important feature for many applications, there is always a trade-off between regularity and object-boundary adherence.

Additionally, recent advances in superpixel algorithms allow the usage of object information (*e.g.* saliency maps) to improve segmentation and provide control over the behavior of superpixels [5, 6]. To the best of our knowledge, no saliency estimator has explored a saliency-based superpixel segmentation yet.

3.6 Conclusion

The available unsupervised saliency estimators usually model saliency using a combination of bottom-up image-extracted information and top-down observation-based saliency models. The bottom-up information extraction strategies and the top-down saliency models depend on assumptions made based on domain-specific knowledge.

By using these pre-selected assumptions as an integral element of their algorithms, the available saliency estimators provide off-the-shelf solutions that often achieve satisfactory results on the scenarios they were proposed to perform on; however, extending these methods to other domains is unfeasible due to the presumed salient characteristics hardly integrated into the method. In this regard, there is a need for a flexible saliency estimation method that can be easily extended to other image domains, allowing for on-demand

addition of problem-fitted assumptions.

Chapter 4

Iterative Saliency Estimation fLexible Framework

The *ITerative Saliency Estimation fLexible Framework* (ITSELF) is a graph-based algorithm that leverages domain knowledge and low-level image information to estimate and enhance object-based superpixels and saliency iteratively. The interaction between superpixel-based saliency and saliency-based superpixels allows for an iterative enhancement cycle that characterizes the core of ITSELF (Figure 4.1). The framework's flexibility comes from user-defined query-selection strategies and prior map modeling. Queries define examples of foreground/background regions to be compared to, while the priors enhance the initial estimation.

On Sections 4.5 and 4.4 we present example implementations of these elements. Lastly, we show how they are integrated to be used on ITSELF (Section 4.2).

4.1 Object-based Superpixel Segmentation

Superpixel segmentation is the process of partitioning an image into n connected regions of pixels (*i.e.* superpixels) that share similar characteristics. Superpixels are used mainly in saliency estimation for reducing the algorithm's input and provide better object boundary definition. However, when it comes to saliency estimators, the choice of which superpixel segmentation algorithm to use is rather overlooked.

Recent studies have allowed the incorporation of object information (*e.g.* saliency) into superpixel segmentation [5, 6]. To the best of our knowledge, the only method that leverages saliency maps for superpixel segmentation is the Object-based Iterative Spanning Forest (*OISF*) [5], an extension of the *Iterative Spanning Forest* [48] framework. *OISF* is composed of three main steps: (i) estimate a representative pixel for each superpixel (namely seeds); (ii) form pixel groups according to how strongly connected they are to the seeds; (iii) recompute the seeds. Through n_r iterations, the segmentation result is improved through subsequent executions of steps (ii) and (iii).

For seed estimation, *Belem et.al.* [6] proposed two approaches that takes object information into consideration, Object-based grid (OGRID) and Object Saliency Map sampling by Ordered Extraction (OSMOX). On both strategies, the user can control the

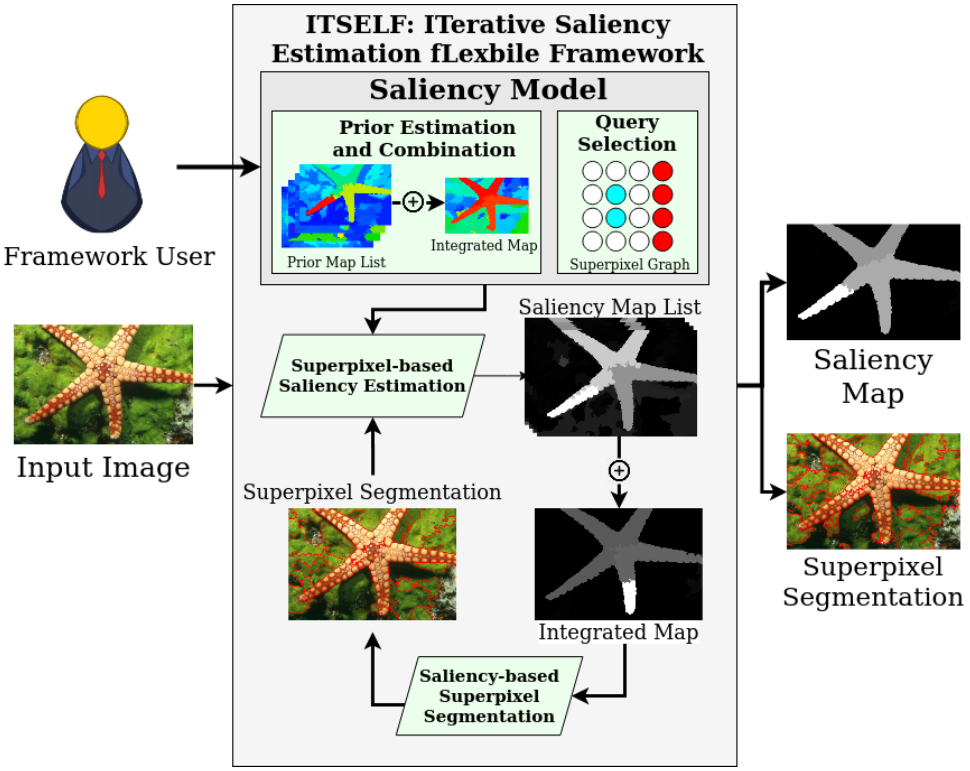


Figure 4.1: Detailed ITSELF's overview. The framework user can define specific saliency models according to prior knowledge and query selection strategies. Note that the saliency and prior map integration is depicted as a plus sign.

number of superpixels and the percentage of object seeds $\rho \in [0..1]$ — *i.e.* how many seeds will fall into salient regions (Figure 2.5). However, OGRID loses saliency information by requiring a thresholded map when determining the salient regions. On the other hand, OSMOX is faster, has equivalent results, and takes advantage of the saliency map's nuances.

Let $n_o = n\rho$ and denote the number of object seeds. Briefly explaining OSMOX, n_o seeds are selected from a priority queue of pixels, where the priority is defined according to the saliency of the pixel's neighbors. To ensure better seed distribution, for each pixel selected as object seed, the priority of adjacent pixels is reduced, and the priority queue is rearranged. The previous steps are repeated until the number of seeds is obtained — it is analogous for $n - n_o$ background seeds. A more detailed explaining was covered in Section 2.5.

With the seeds selected, *OISF* runs the *Image Foresting Transform* (*IFT*) algorithm [18] for delineating the superpixels. The *IFT* computes an optimum-path forest, where the seeds are the roots of the trees, and optimality is defined in terms of a path-cost function. Non-seed pixels are aggregated to the tree that provides the minimum path-cost to it. Each tree of the resulting forest is taken as a superpixel.

OISF proposes the path cost to be additive, where the added value is derived from the color and the saliency difference between pixels (Equation 2.12. Even though OISF's path-cost function may not satisfy some conditions to achieve optimality [14], the resulting trees are suitable for image representation.

After the first segmentation is finished, the results can be improved by recomputing the seeds and running another iteration of the pipeline. In this work, the seeds are repositioned using the strategy proposed on [5] (*i.e.* the superpixel medoid on the feature space). Additionally, by enhancing the saliency map over iterations of *ITSELF*, the next *OISF* segmentation is being performed on an improved initial set of seeds.

At each ITSELF iteration t, the number of superpixels may change in order to compute saliency on multiple scales. For that, we added a parameter $\kappa \in (0..1]$ that redefines the number of superpixels on each iteration: $n^{t+1} = n^t\kappa$; in which $n^1 = n$.

4.2 Superpixel-based Saliency Estimation

Let $\mathcal{S}$ be the set of all superpixels, $S, R \in \mathcal{S}$, and $Q \subset \mathcal{S}$ be the subset of query superpixels. We start by representing the image as a superpixel weighted graph $\mathcal{G} = (\mathcal{S}, E)$, as described on Section 2.

To allow for the user to control the importance of query over non-query edges, the edges start with parametrically defined weight $\boldsymbol{e}(S, R)$, where query edges have weight $\psi \in [0..1]$ and adjacency edges have weight $1 - \psi$.

Similar to Cheng *et al.* [13], we define dissimilarity in terms of color differences between superpixels. Let C_I be the set of all unique colors that compose I, $C_S \subseteq C_I$ be the existing colors in a superpixel S, and $p(c, S), c \in C_S$ be the percentage of c colored pixels on S. We incorporate the dissimilarity measure to the graph by updating the edge weights using a Gaussian function:

$$\boldsymbol{e'}(S, R) = \boldsymbol{e}(S, R) \sum_{\forall c_i \in C_S} \sum_{\forall c_j \in C_R} \exp^{-\frac{\|c_i - c_j\|}{\sigma_s}} p(c_i, S)p(c_j, R), \tag{4.1}$$

where $\sigma_s \in (0, 1]$ is the variance and controls the rate in which the distance function increases, and $(S, R) \in E$.

Then, let $E_F \subset E_Q$ be the subset of foreground-query edges. We invert the foreground query weights to account for similarity instead dissimilarity, defining *vertex saliency* to be:

$$\boldsymbol{VS}(S) = \sum_{\forall R \in E \setminus E_F} \boldsymbol{e'}(S, R) + \sum_{\forall F \in E_F} 1 - \boldsymbol{e'}(S, F). \tag{4.2}$$

Finally, we incorporate the prior domain information simply by multiplying the saliency score of each vertex by the *normalized combined prior map* $\boldsymbol{PS}$ (detailed in Subsection 4.3) to get the final saliency score:

$$\boldsymbol{S}(S) = \boldsymbol{VS}(S)\boldsymbol{PS}(S) \tag{4.3}$$

The resulting saliency image maps to each pixel p the saliency value of its corresponding superpixel.

Additionally, instead of taking the last iteration result as final, we used the prior integration step to combine the multiple computed maps. We only discard the first estimated map as it often highlights a big part of the background.

4.3 Prior and Saliency Map Integration

Prior knowledge of how humans perceive saliency allows for assumptions to be drawn on which characteristics are determinant when defining saliency. These assumptions alone are often insufficient to accurately identify salient regions. However, by combining multiple priors, it is possible to create more accurate models (Figure 1.5).

We propose ITSELF to be flexible to the number of priors incorporated into the model without over increasing the relevance of top-down information over bottom-up strategies. For such, it requires a combination strategy that allows any number of prior maps to be combined into a single map. The resulting prior map is then used during the saliency estimation step (Equation 4.3).

The integration is done using the cellular automata method proposed by Qin *et al.* [41] as described in Section 2.6. The main difference between the version implemented within ITSELF is that we allow the automata to aggregate new cells (through newly computed maps) during the process of updating. By doing so, the automata is used for integrating different visual models, and is also used to improve the result of the individual models by aggregating better estimations over better delineated superpixels. Additionally, by changing the number of superpixels at each of ITSELF's iterations, the prior maps created are being computed on different scales. Thus, the automata serves three purposes on the prior models: add multi-scale information; improve individual models over-time with increasingly better delineated superpixels; integrate all priors into a single map.

We also use cellular automata to combine the output saliency maps from each of ITSELF's iterations. We realized early on our experiments that on later iterations of ITSELF, the saliency score of less salient parts of complex object would start to be confused as background. By using the cellular automaton and considering the results of previous iterations, the results of later iterations are more consistent 4.2. Additionally, similar to the prior maps, using the automaton allows for ITSELF to account for multiple scales if the number of superpixels used on each iteration is different.

4.4 Prior Modeling

In this section, we present the models we implemented for each prior used in our experiments. The prior maps are represented the same way as a saliency map, however, to easily differentiate between both, the prior maps on this document are represented as heat maps where lower values are represented on cold colors(blue→green) and higher values on hot colors(yellow→red).

4.4.1 Center-surround prior

A widespread assumption for natural images is that the salient object will be near the center of the image [13, 44]. In this regard, let $p_c \in P$ be the center pixel of the image. The center distance of a superpixel to the image center is defined as: $\boldsymbol{CD}(S) = \frac{1}{|S|} \sum_{\forall q \in S} \|q - p_c\|$.

However, to change the increase rate of the distance function, the values are put into a Gaussian centered on p_c, thus, the center prior score is defined as follows:

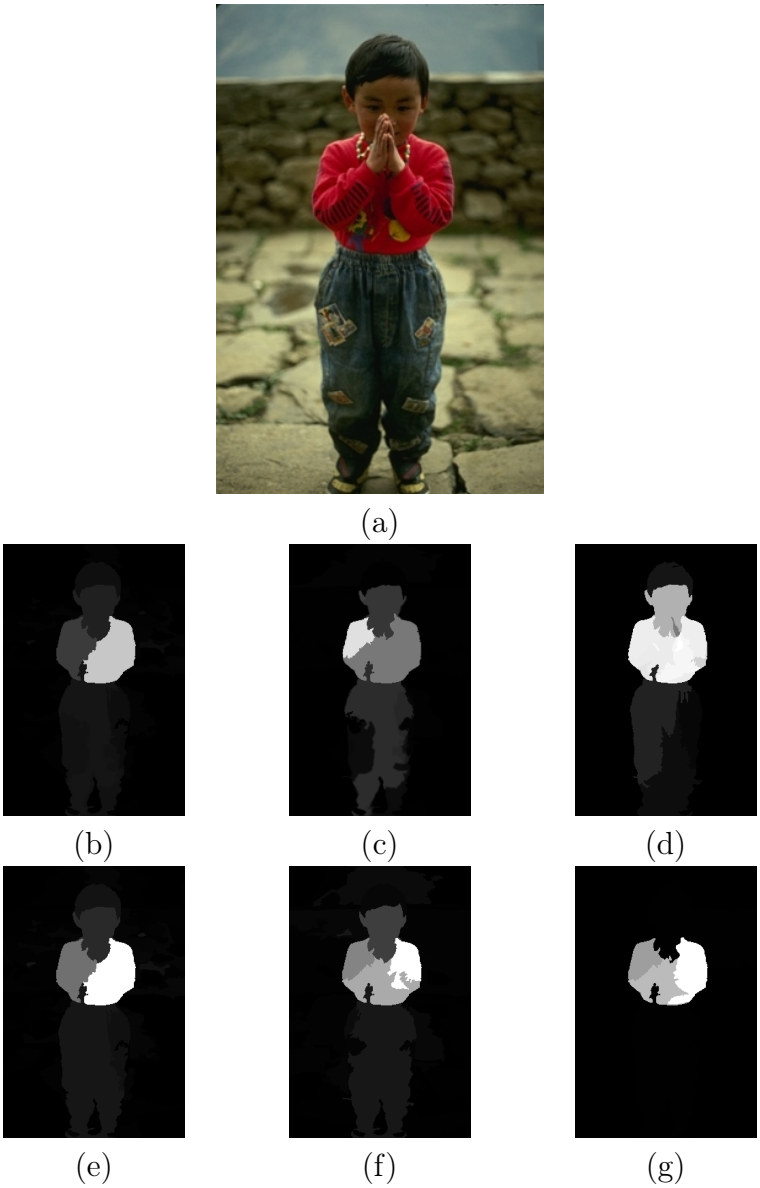


Figure 4.2: (a) Input image. (b-d) The result of ITSELF on iterations one, five and eight (final) using the automaton; (e-g) The result of ITSELF on the same iterations as (b-d) but without using the automaton to take previous iterations into consideration.

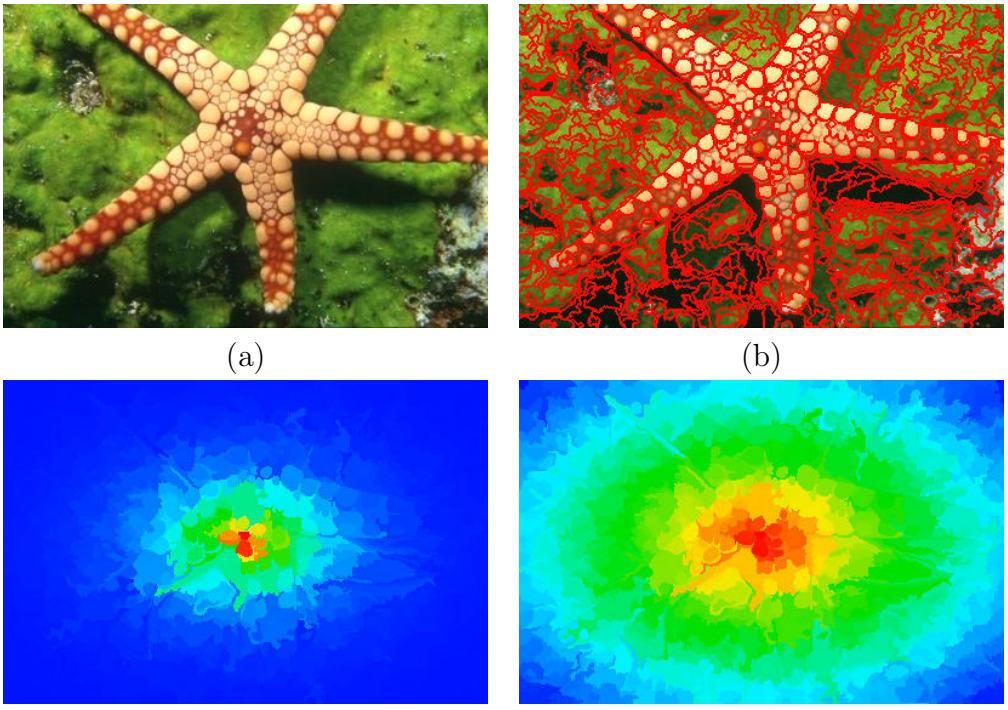
$$\boldsymbol{CP}(S) = \exp^{-\frac{\boldsymbol{CD(S)}}{\sigma_1^2}} \tag{4.4}$$

Smaller values of $\sigma_1 \in (0..1)$ causes superpixel farthest to center to be less relevant (Figure 4.3).

4.4.2 Global color uniqueness prior

The saliency score represents how much an object stands-out in a scene. A defining characteristic when estimating said score is color uniqueness. Colors that appear the least are rarer in the image and may stand out [13, 27].

Let C_I be the set of all unique colors that compose I, $P_c \subset P$ be the set of all pixels



(a) (b) (c) (d)

Figure 4.3: (a) Original image. (b) Superpixel Segmentation; (c) and (d) Center prior maps with $\sigma_1 = 0.1$ and $\sigma_1 = 0.9$, respectively.

of color c $in C_I$, and $\boldsymbol{p}(c) = \frac{|P_c|}{|P|}$. Similar to the center prior, we use a Gaussian function to control the increasing rate of the the distance measure. In this regard, we define the *Color Uniqueness Score* as $\boldsymbol{US}(c) = \exp^{\frac{p(c)}{\sigma_2^2}}$.

However, even after quantization, there are several similar colors (*e.g.* slightly different tones of the same color), creating artifacts counter-intuitive to the human perspective. To reduce the impact of similar color uniqueness, similar to Cheng *et al.* [13], we smooth the uniqueness score based on the average uniqueness of similar colors. For every pair $(c_i, c_j), c_i \neq c_j$, we define the color similarity weight to be $\boldsymbol{ws}(c_i, c_j) = \exp^{-\frac{\|c_i, c_j\|}{\sigma_2^2}}$, and prose the final global color-uniqueness score to be:

$$\boldsymbol{US}'(c_i) = \sum_{\forall c_j \in C_I} \boldsymbol{US}(c_j)\boldsymbol{ws}(c_i, c_j) \tag{4.5}$$

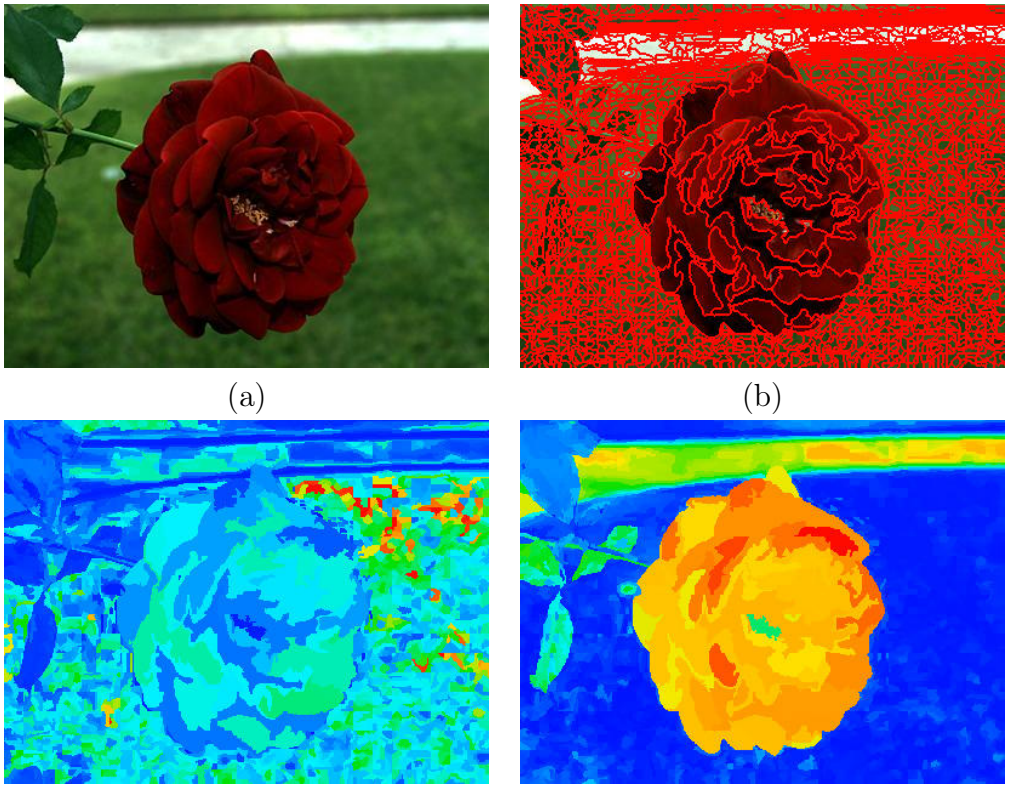
Figure 4.4 depicts the improvement when smoothness is applied.

Each superpixel is then assigned a value according to the colors of the pixels that composes it: $\boldsymbol{GP}(S) = \frac{1}{\|C_S\|} \sum_{\forall c \in C_S} p(c, S)\boldsymbol{US}'(c)$.

4.4.3 Color-based priors

Based on observation of the Human Visual System, a common assumption is that red and yellow tones are naturally salient.

Identifying red and yellow colors is straightforward in the **L*a*b*** colorspace: higher values on the **a** channel describe red tones,while high values on the b channel, yellow.



(a) (b) (c) (d)

Figure 4.4: (a) Original image. (b) Superpixel Segmentation; (c) and (d) Global color-contrast prior maps without and with the smoothness step, respectively. Note how slight changes in tones of green impact negatively the method without smoothness.

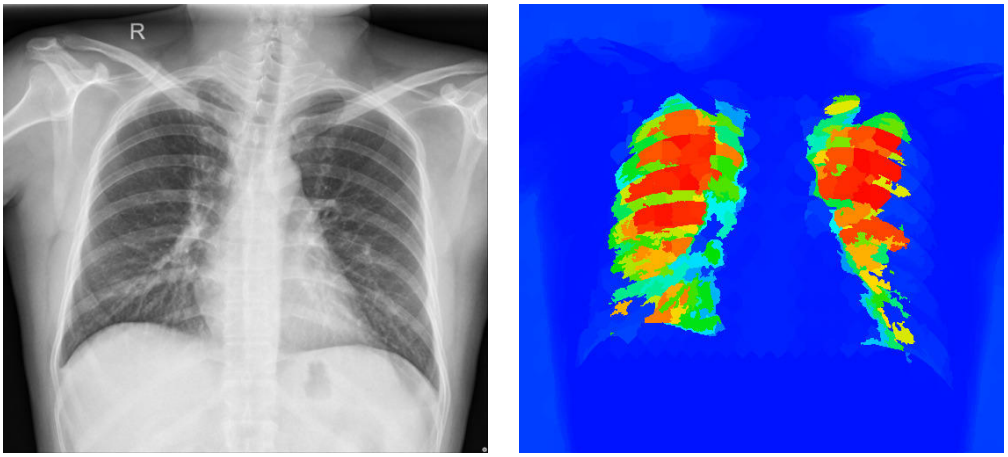
Therefore, we define a red/yellow score $\boldsymbol{RY}(c)$ to be the sum of its a and b channels. As in the previous priors, we use a Guassian function to exert control over the functions increase rate, redefining the score to be $RY'(c) = \exp^{\boldsymbol{RY}(c)/\sigma_3^2}$.

Lastly, we propagate the color values to every superpixel using a weighted average:

$$\boldsymbol{RP'}(S) = \sum_{\forall c \in C_S} p(c, S)\boldsymbol{RY'}(c)) \tag{4.6}$$

Although red and yellow are the most naturally salient colors, the same algorithm can be applied to other colors when required for specific objects. As an example, we know that on x-ray images of the thorax, the lungs are often darker than the other structures. So, we implemented a color prior that highlights black regions (Figure 4.5).

All color-based priors follow the same principle of adding or subtracting the **L*a*b*** channels: white and black requires **a** and **b** to be closer to 0, with white having higher values on the **L** channel; the more negative the value of the **a** channel, the greener the color tone is, and the same goes for blue on the **b** channel. Changing the operations done on the channels yields new color priors.



(a) (b)

Figure 4.5: (a) Original image with object scribbles in light-blue. (b) Black prior map with $\sigma_3 = 0.5$. Additionally, we reduced the saliency of black regions connected to the image boundaries because of the natural color of the xray plate.

4.4.4 Saliency-based priors

Due to the iterative nature of *ITSELF*, we created a method that extrapolates a prior map from a previously computed saliency map, trying to reduce spurious saliency values of background regions sharing non-salient colors.

We propose a color-saliency prior that attributes a saliency value to a superpixel depending on how globally salient its colors are (Figure 4.6).

The global saliency of a color is defined as:

$$\boldsymbol{SC}(c) = \frac{1}{|P_c|} \sum_{\forall p \in P_c} \boldsymbol{S}(p) \tag{4.7}$$

Let C_I be the set of all unique colors that compose I, $C_S \subset C_I$ be the subset of colors that compose a superpixel S. We then compute the saliency prior score to each superpixel, combining their color scores:

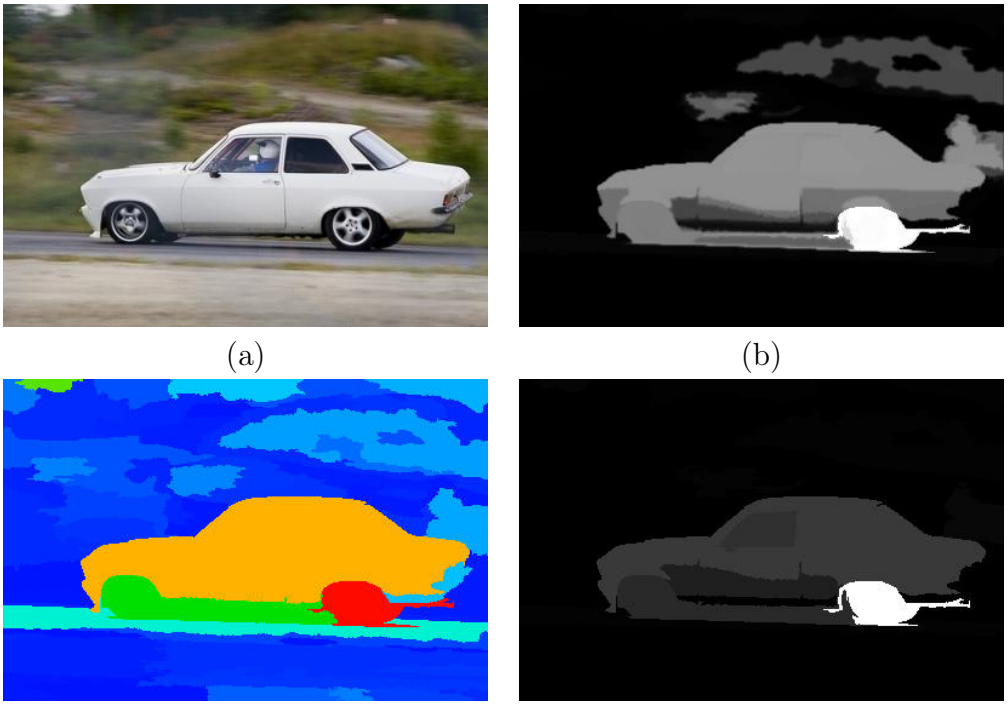
$$\boldsymbol{CS}(S) = \frac{1}{|C_S|} \sum_{\forall c \in C_S} \boldsymbol{SC}(c) \tag{4.8}$$

Despite only presenting a color-based saliency prior, other features (*e.g.* texture, shape, or size) could be used to create new priors similarly.

4.4.5 Focus prior

One could draw a natural correlation between focus and saliency. When observing a picture with different focal points, our eyes tend to naturally ignore blurred regions prioritizing focused ones. Accordingly, identifying focused regions can improve the saliency estimation task [27].

As presented by Jiang *et.al.* [27], the focus of a region is closely related to its degree of blur. With blurriness being the lack of sharply defined edges, *focusness* is more easily



(a) (b) (c) (d)

Figure 4.6: (a) Original image. (b) Saliency map before multiplying by the proposed color-saliency based prior; (c) the color-saliency based prior derived from (b); (d) the result of multiplying the initial saliency with the proposed prior. Note the error reduction on background saliency.

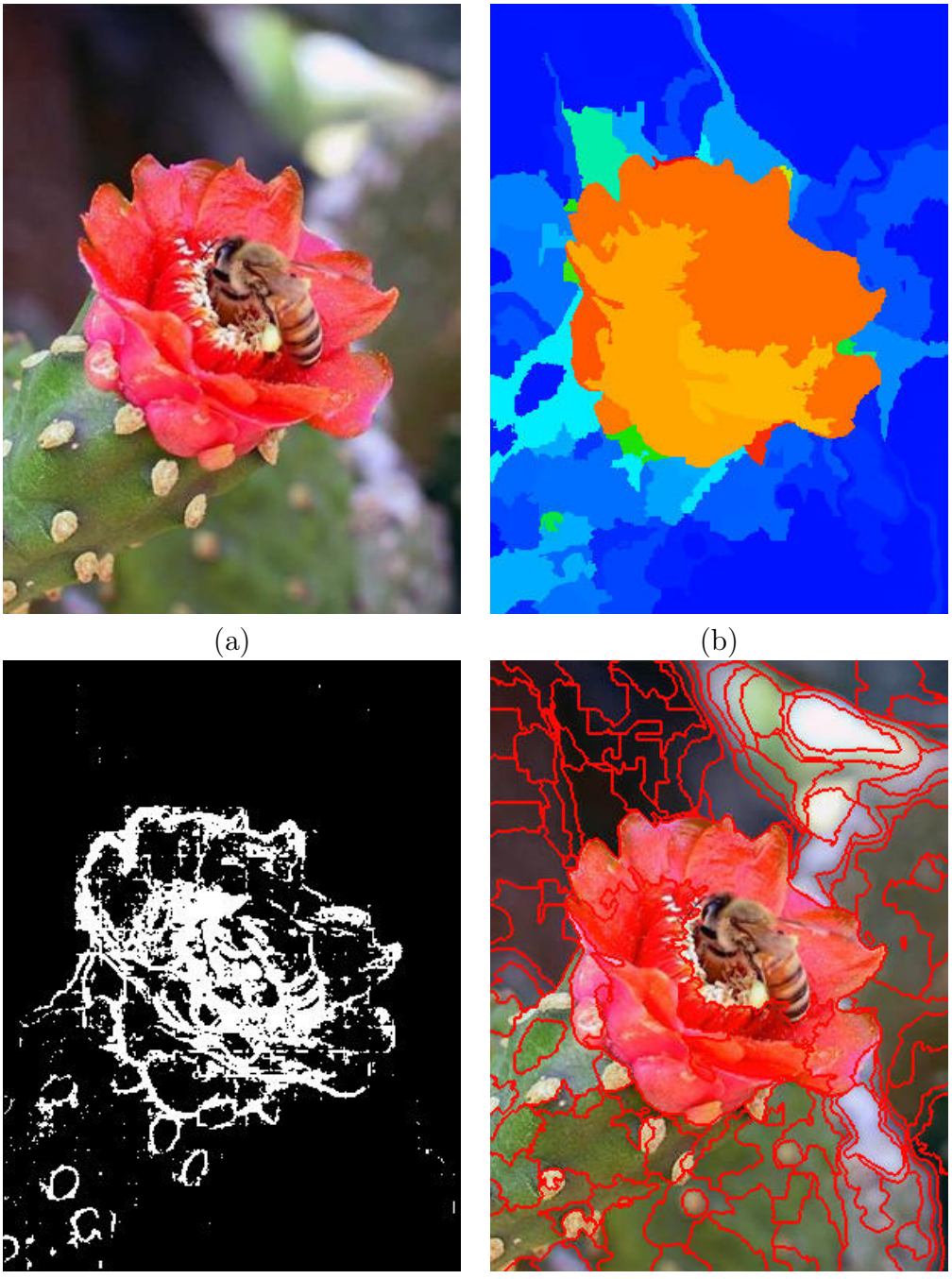
quantifiable by looking at the edges of objects rather than their interior. Consequently, the first step when computing focusness is to identify object edges on the image. There are several edge detection algorithms available in the literature, however, we opted on using a simple thresholded gradient image. Let the gradient $\nabla(p) = \|\boldsymbol{I}(p), \boldsymbol{I}(q)\| \forall q \in \mathcal{A}_4$, and $P_e \subset P$ be the set of edge pixels. We consider that $p \in P_e \leftrightarrow \nabla(p) > \omega$, where ω is the Otsu threshold [38] of $\mathcal{I}$.

Within *ITSELF*, the regions are delimited by superpixels and, thus, the focusness score can be defined by correlating superpixels to the detected edges. Superpixel segmentation also uses gradient information, and the created superpixel boundaries are frequently located in regions with a higher gradient. However, in blurred regions, the natural image edges will not exceed the threshold and will not be present on the estimated edges. In this regard, focused regions should have a higher match between superpixel boundaries and sharp image edges (Figure 4.7).

Let $P_b \subset S$, $p_b \in P_b$ be a boundary pixels of S — *i.e.* $\exists q \in \{\mathcal{A}_8(p_b), R\}, R \neq S$. We define the focusness score of S by:

$$\boldsymbol{FS}(S) = \frac{|P_b \cap P_e|}{|P_b|} \tag{4.9}$$

Like the other priors, we map the focusness score to its location within a Gaussian function:



(a) (b) (c) (d)

Figure 4.7: (a) Original image. (b) Result of the focus prior. (c) Estimated object edges; (d) Object-based superpixel segmentation.

$$\boldsymbol{FP}(S) = 1 - \exp^{-\frac{\boldsymbol{FS}(S)}{\sigma_4^2}} \tag{4.10}$$

4.4.6 Ellipse-matching prior

Thanks to OISF's capability of representing objects with few superpixels, shape-based priors are viable. As an example, we created a prior that highlights elliptical objects to increase ITSELF's precision on an in-house dataset of intestinal parasite eggs (Section

5.1).

To score how elliptical the superpixels are, we first compute a Tensor Scale Representation (*TSR*) of each of them. The *TSR* of a homogeneous region is a parametric representation of the best fit ellipse enclosed inside the region. Each ellipse is defined through its orientation (the angle between the ellipse's major axis and the image's y-axis), its anisotropy (the ratio between its major and minor axis), and thickness (size of the minor-axis). To compute the *TSR* for every superpixel, we use a slightly modified version of an optimized algorithm [37]. The algorithm consists of identifying the edges of the homogeneous regions, finding the orientation of the best-fit ellipse, and computing the length of the ellipse's semi-axis.

The main difference of our implementation when compared to Miranda's is when defining the region edges. Let $p \in S, q \in R, S \neq R$, and $P_e \subset P$ be the set of all region edges. We consider that $p \in P_e \leftrightarrow \exists q \in \mathcal{A}_8(p)$.

The last two stages are implemented as described in [37], taking the superpixels as the homogeneous regions and the center pixel of the superpixel as the center of the ellipse.

Afterwards, we estimate an ellipse matching for each superpixel (Figure 4.8):

$$\boldsymbol{EM}(S) = \frac{1}{|S|} \sum_{\forall p \in S} \delta_e([\|p, f_1\| + \|p, f_2\|] < 2l), \tag{4.11}$$

where $\delta_e(\cdot) = \{0, 1\}$ determines whether a pixel is positioned inside its respective ellipse, and f_i are the estimated *foci*.

The final Ellipse-matching prior is also weighted by a Gaussian and is computed as follows:

$$\boldsymbol{EP}(S) = 1 - \exp^{-\frac{\boldsymbol{ES}(S)}{\sigma_5^2}} \tag{4.12}$$

Specific to the parasite dataset, we improve the ellipse prior result by adding a size constraint:

$$\boldsymbol{EP'}(S) \;=\; \begin{cases} \boldsymbol{EP}(S) & \text{if } |S| \in (s_0, s_1), \\ min(\boldsymbol{EP}(\mathcal{S})) & \text{otherwise,} \end{cases} \tag{4.13}$$

where s_0 and s_1 are, respectively, the lower and upper limits of the size range defined empirically. Figure 4.9 shows the improvement achieved by size filtering.

4.4.7 Scribbles based priors

Object saliency maps have been frequently used to assist interactive object segmentation [17, 55, 45]. In this scenario, the objects' locations are given by the user who interactively places scribbles in the object and background.

These user placed scribbles can be used as precise object detection, allowing the creation of several new priors with high accuracy. As a simple example, we can create location priors (similar to *center-surround*) regarding the detected objects: A point has brighter values when they are close to object scribbles or far from background ones. Figure 4.10

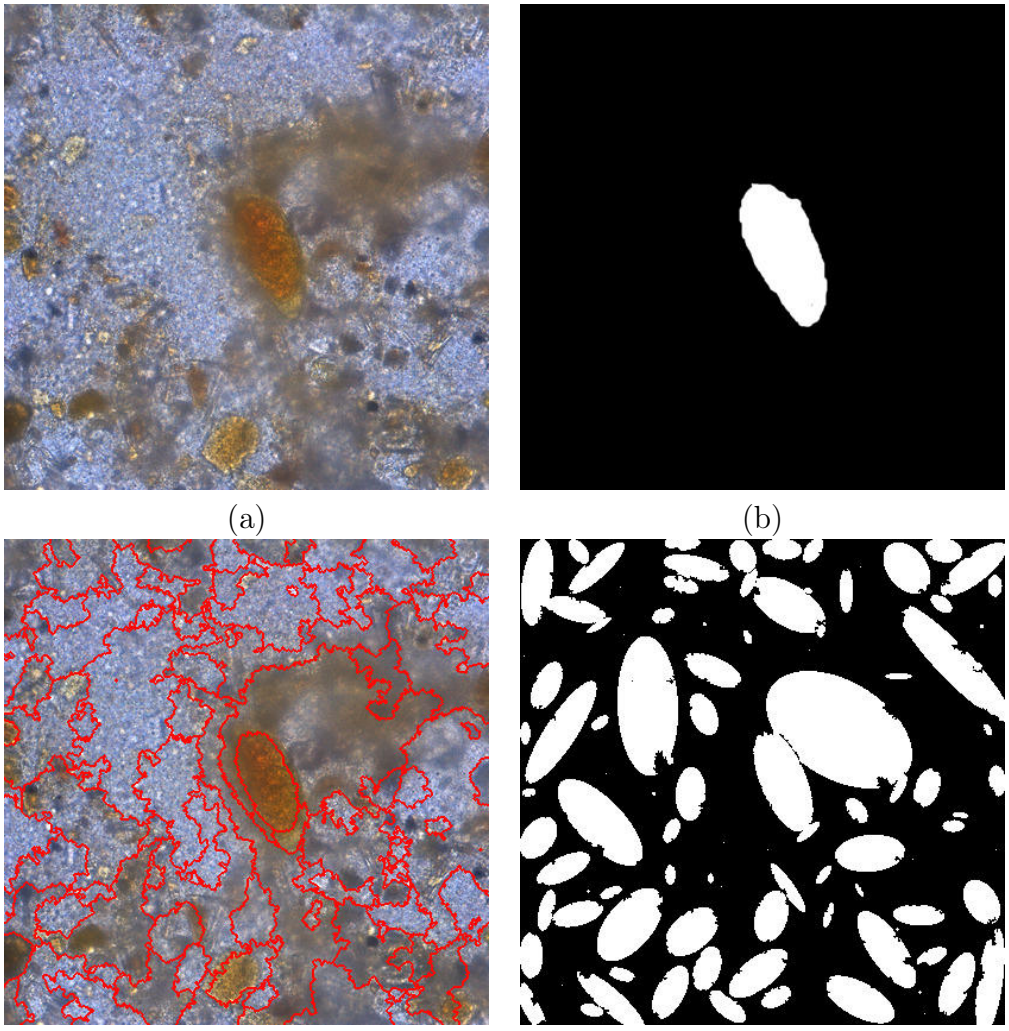


Figure 4.8: (a) Original image. (b) Object mask of the parasite. (c) Superpixel segmentation; (d) Ellipse Matching of each superpixel

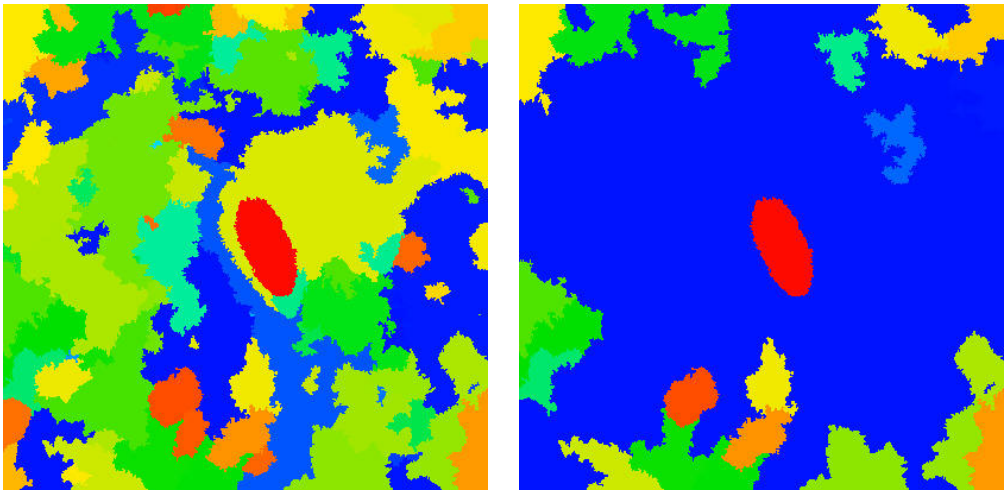
shows the result of using object scribbles as a location prior.

It is worth noting that scribble based location priors can be used, for instance, segmentation. The challenge in instance-segmentation is to individually segment objects of the same class in a picture with multiple objects. Take Figure 4.11 as an example: There are multiple flamingos on the image, but one may only interested in the top right flamingo. To fulfill such needs, we use the location scribble-based prior, reducing the saliency of all other objects that are not close to the user-provided marker.

Note that other highly accurate priors could be created by using scribbles. They could be based on color, texture, or even shape and size by exploring object-based superpixels. We strongly advise exploring these possibilities in further works.

4.5 Query Selection

We propose three different approaches to estimate queries: (A) border-based query, assuming most of the image boundary regions are background on natural images; (B) saliency-



(a) (b)

Figure 4.9: (a) Ellipse-based prior without size filtering (b) Result of reducing region saliency by size.

(a) (b)

Figure 4.10: (a) Original image with object scribbles in light-blue. (b) Scribble-based location prior map.

based, used to incorporates any pre-computed saliency map into the framework.

4.5.1 Border-based Query Selection

We propose combining both boundary connectivity [63] and multi-map estimation [59, 39, 60] to further reduce the miss-selection of background regions. For such, instead of using the four sides of the image as the multiple maps, we propose clustering the superpixels based on their color similarity. We then compute a saliency map for each of the clusters

(a) (b)

Figure 4.11: (a) Original image with object scribbles in light-blue. (b) Scribble-based location prior map.

that contain at least one superpixel on the image boundary. During this computation, we only use the cluster's superpixels touching the image boundaries as queries.

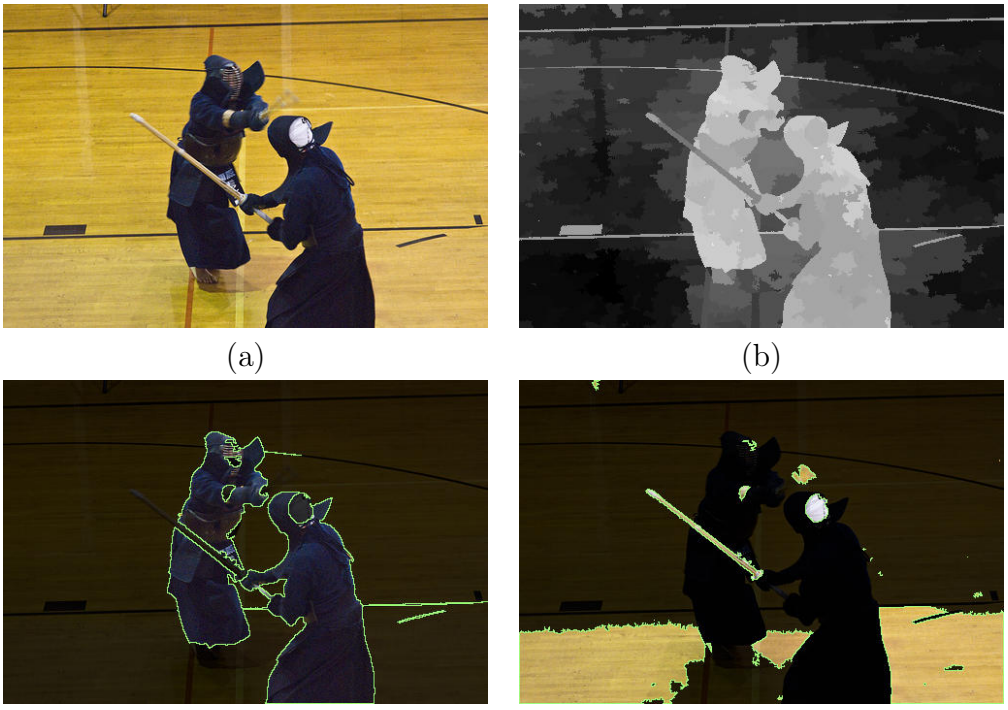
Any clustering algorithm could be used; however, we opted on using the *Unsupervised Optimum-Path Forest (OPF)* [43]: A graph-based algorithm that performs clustering by solving an optimum-path forest on a graph of samples. OPF finds an adequate number of clusters g automatically; therefore, different images may have a different number of clusters.

Let $\mathcal{S}_g \subset \mathcal{S}$ be the set of superpixels contained in cluster g, and $\mathcal{B}_g \subset \mathcal{S}_g$ be the set of superpixels in $\mathcal{S}_g$ that touches the image borders. For each cluster, we compute a saliency map $\boldsymbol{SM}_g$ using Equation 4.3 and a boundary-connectivity score w_g. In this work, the *boundary connectivity score* of cluster measures how many of its superpixels touch the image border, and is defined as $w_g = |\mathcal{B}_g|/|\mathcal{S}_g|$.

We then perform a weighted average to attribute to each superpixel a single saliency score:

$$\boldsymbol{CS}(S) = \frac{1}{W} \sum_{g}^{n_g} w_g SM_g(S) \tag{4.14}$$

where $W = \sum_{g}^{n_g} w_g$. With the saliency score of each superpixel, the final saliency map is the propagation of the saliency scores of each superpixel to every pixel that composes it. A visual representation of the combination of clustering and the boundary-connectivity score is depicted in Figure 4.12.



(a) (b)

(c) (d)

Figure 4.12: (a) Input image. (b) The result combination of the boundary clusters; (c) The highest boundary-connectivity score cluster with $w_c = 0.453$; (d) A boundary cluster containing most of the object with boundary-connectivity score $w_c = 0.142$. Note that the combined saliency map is not the final result of ITSELF, rather it is the simple combination of the boundary clusters.

4.5.2 Saliency-based Query Selection

Queries are subsets of image regions that are representative when estimating saliency, given a set of predicates. Whether the queries are good representations of foreground or background, the importance of such regions can be encoded by a saliency map. Thus, given a saliency map $\boldsymbol{SS}$ and a threshold μ, a superpixel is selected as a query if its saliency value exceeds the threshold.

Chapter 5

Experiments and Results

We compare ITSELF to two other popular saliency estimators, namely the *DiscriminativeRegional Feature Integration Approach* (DRFI) and the *Structured Matrix Decomposition* (SMD). Assuming the background to be more homogeneous than the foreground usually, SMD uses the low-rank (LR) matrix theory to approximate the redundant background regions on a low-rank feature matrix, while a sparse sensory matrix represents sparse salient object regions. They use connectivity constraints, and a regularization step used to assist images with a cluttered background. Additionally, SMD incorporates location, color, and background priors to improve their results further.

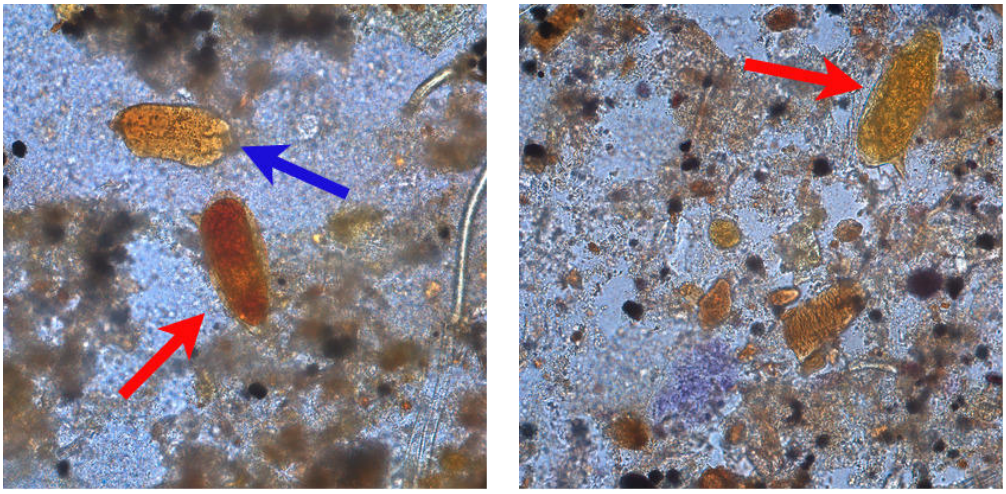
Despite DRFI being a supervised algorithm, it uses only hand-crafted features extracted from the input image, using a Random Forest to combine them and form the saliency score. The features are similar to other unsupervised methods (color, texture, and guess location). They compute multiple saliency scores on multiple scales and combine them at a fusion step. By learning the importance of each feature for different datasets, DRFI has the potential to be more easily extensible to other image domains.

We did not include comparisons to the state-of-the-art graph-based algorithms because there was no code available, and we could not run the same experiments and evaluate on the same metrics as we did the others. However, they have the same inflexibility of the other methods, incorporating pre-selected assumptions into their models.

5.1 Datasets

To validate our method, we used four popular natural image datasets: the **MSRA10K** [33], which is the largest dataset selected (10000 images) and is composed of images with a singular salient object and a somewhat simple background; the **ECSSD** dataset [53], containing 1000 images of a singular salient object in a complex background; the **DUT-OMRON** dataset [54], which was proposed to n saliency detection dataset, composed of 5,168 complex images containing one or more salient objects; and **ICoSeg** [4], which is composed of 643 images, most of them containing multiple salient objects.

Additionally, we ran experiments on an in-house biomedical image dataset of intestinal **parasite eggs**. The dataset is composed of 630 images of *schistosoma-mansoni* eggs obtained via TF-test [21], with. The background is overloaded with fecal impurities that



(a) (b)

Figure 5.1: (a) A parasite egg (red arrow) and a fecal impurity (blue arrow) that shares similar characteristics to the eggs; (b) A heavily cluttered image with one parasite egg (red arrow)

share similar characteristics to the eggs, posing a challenge to highlight the wanted objects alone (Figure 5.1). Differently than the impurities, the eggs are elliptical and fall into a specific size range.

Lastly, we used a **lung x-ray** dataset proposed in a Kaggle segmentation challenge to showcase that ITSELF can be extended to grayscale medical images. This dataset is composed of 704 images and contains normal and abnormal x-rays with manifestations of tuberculosis. It is required to attribute the data source to the National Library of Medicine, National Institutes of Health, Bethesda, MD, USA, and to the Shenzhen No.3 People's Hospital, Guangdong Medical College, Shenzhen, China. This dataset was made viable thanks to [25, 9].

5.2 Parameter tuning and Experimental setup

For optimizing the methods' parameters we created subsets of size $N = min(\frac{\|D\|}{10}, 100)$ where $|D|$ is the dataset size. We want to make sure that the user does not need many images to achieve satisfactory results, so N limits the MSRA10K and DUT-OMRON training set size.

Regarding the parameters, some parameters were fixed, and others changed for each dataset. The dataset-specific parameters values are grouped on Table 5.1. The fixed parameter values will be presented as we list them.

For superpixel segmentation, there are six parameters: the number of superpixels n; the number of foreground seeds $n_o = 3$; the number of OISF iterations over recomputed seeds $\hat{t}$, the superpixel size regularity ($\alpha = 0.8$), the border adherence weight ($\beta = 12$) and the saliency weight (γ'). On the saliency computation, there are two parameters: the query region importance ψ and the saliency variance ($\sigma_s = 0.4$). For the prior integration step, there are also two parameters: the number of iterations $t' = 1$; and the

update-strength Λ. Although $t' = 1$, the automaton updates over the number of ITSELF iterations.

We used two different types of color priors, one that highlights red/yellow colors and another to highlight color intensities. The intensity prior was used on all natural-image datasets as well as on the x-ray dataset; however, on the x-ray, we highlight darker intensities.

For prior estimation, there are eight parameters: the variance of each prior variance ($\sigma_i \| i \in (1..6)$ — where σ_3 is related to the red/yellow prior and σ'_3 to intensities; and the size constrains for the ellipse prior $s_0 = 1500$, $s_1 = 5000$. Lastly, we run all the experiments using $i = 8$ full ITSELF iterations.

Regarding SMD, they proposed a method to be used without any training step. In this regard, we used their available code without any modifications or parameter tuning. Note that SMD did a pre-training on the used datasets but are not clear regarding the size of their training split. As for DRFI, we used their available implementation and the same splits as we did for ITSELF.

	ECSSD	DUT_OMRON	ICOSEG	MSRA10K	Lungs	Parasites
σ_1	0.2	0.2	0.2	0.2	—	—
σ_2	0.2	0.5	0.5	0.5	—	0.2
σ_3	0.2	—	0.2	0.8	0.5	—
σ'_3	0.2	0.5	0.8	0.8	—	0.2
σ_4	0.5	0.5	0.5	0.8	0.8	—
σ_5	—	—	—	—	—	1.0
n	200	200	200	200	200	500
γ	2.0	2.0	2.0	1.0	2.0	0.5
Λ	0.01	0.008	0.01	0.01	0.01	0.05
ψ'	0.5	0.3	0.8	0.3	0.3	0.5
$\hat{t}'$	0.5	0.3	0.8	0.3	0.3	0.5

Table 5.1: A list of all parameters values that changed over the datasets.

The query selection strategies employed for each dataset were different from the first iteration of the framework. On further iterations, every dataset used the result of the past iteration to estimate foreground queries. For the natural-image datasets, the first framework iteration uses image-borders background queries to estimate a first saliency map and then uses the result to estimate foreground queries. For the parasites and x-ray datasets, the first iteration uses the combined prior map to estimate foreground queries.

Lastly, when combining the multiple iteration's outputs, we observed that the first saliency estimation is often noisy; thus, we discard it.

5.3 Evaluation Metrics

We used four traditional saliency metrics: weighted F-Measure (WF-Measure); weighted Precision (PRE^ω); weighted Recall (REC^ω); the mean-average error. Moreover, we pro-

pose the usage of boundary recall to quantify this characteristic of the over-salient values of regions close to the object. By increasing the saliency of regions close to the object, the estimated objects, boundaries are moved away from the real object boundaries, reducing the BR (Figure 5.2). The weighted F-Measure is the harmonic mean of PRE^{ω} and REC^{ω}. The PRE^{ω} measures the exactness (*i.e.* whether non-salient regions were defined as salient) and REC^{ω} measures completeness (*i.e.* whether salient regions were defined as non-salient). These metrics were proposed to substitute the traditional binary-image-based precision and recall metrics, removing the need for computing the results on multiple threshold-segmented maps [35]. Rather, the positive and negative ratios are computed based on the difference between a binary map and the saliency probability.

The *mean average error* (MAE) is the mean difference between the saliency map and the ground-truth. Even though the MAE is quite simple, it does not require thresholding the saliency map, not suffering, then, from information loss.

Having well-defined boundaries between object and background is particularly useful on tasks such as weakly-supervised semantic segmentation, where saliency maps may be used as estimates of a pixel-wise mask from an image-level annotation [51] to train more robust algorithms. For that, we use the boundary recall (BR) over saliency maps thresholded by the mean saliency value. BR measures the percentage of match between the estimated object boundaries to the object boundaries in the ground-truth. We consider a boundary tolerance distance of two pixels, as proposed by Achanta *et.al.* [2].

5.4 Natural-image dataset comparisons

As shown in Table 5.2, regarding the four traditional saliency metrics, ITSELF was ranked second on all but one dataset, with SMD getting the best scores. However, not only was SMD pre-trained with an unknown number of images, SMD often highlights non-salient regions close to salient ones.

Comparing the boundary recall of the three methods, ITSELF and DRFI often alternate between first and second place, with SMD always on the bottom.

The saliency/superpixel loop provides a final saliency estimation with more semantic meaning than previous methods. Even though ITSELF was not completely accurate given the ground-truth in Figure 5.2, the wrongfully salient regions are highly different from most of the background or highly similar to the foreground. The WF-Measure of ITSELF and SMD are, respectively, 0.689 and 0.781. Nevertheless, SMD increases the saliency of background regions close to the horse. ITSELF does have a big non-salient region segmented on the top of the image; however, this region does not share similar characteristics to most of the background, unlike SMD's miss-estimation. More examples can be seen on Figure 5.5.

These datasets are composed of highly different objects, and using priors assumptions does make the model prone to error on images that these assumptions fail to describe important salient characteristics (Figure 5.3).

Another scenario where ITSELF often fails is when the salient object is too large or when they are too similar to most of the background (Figure 5.4).

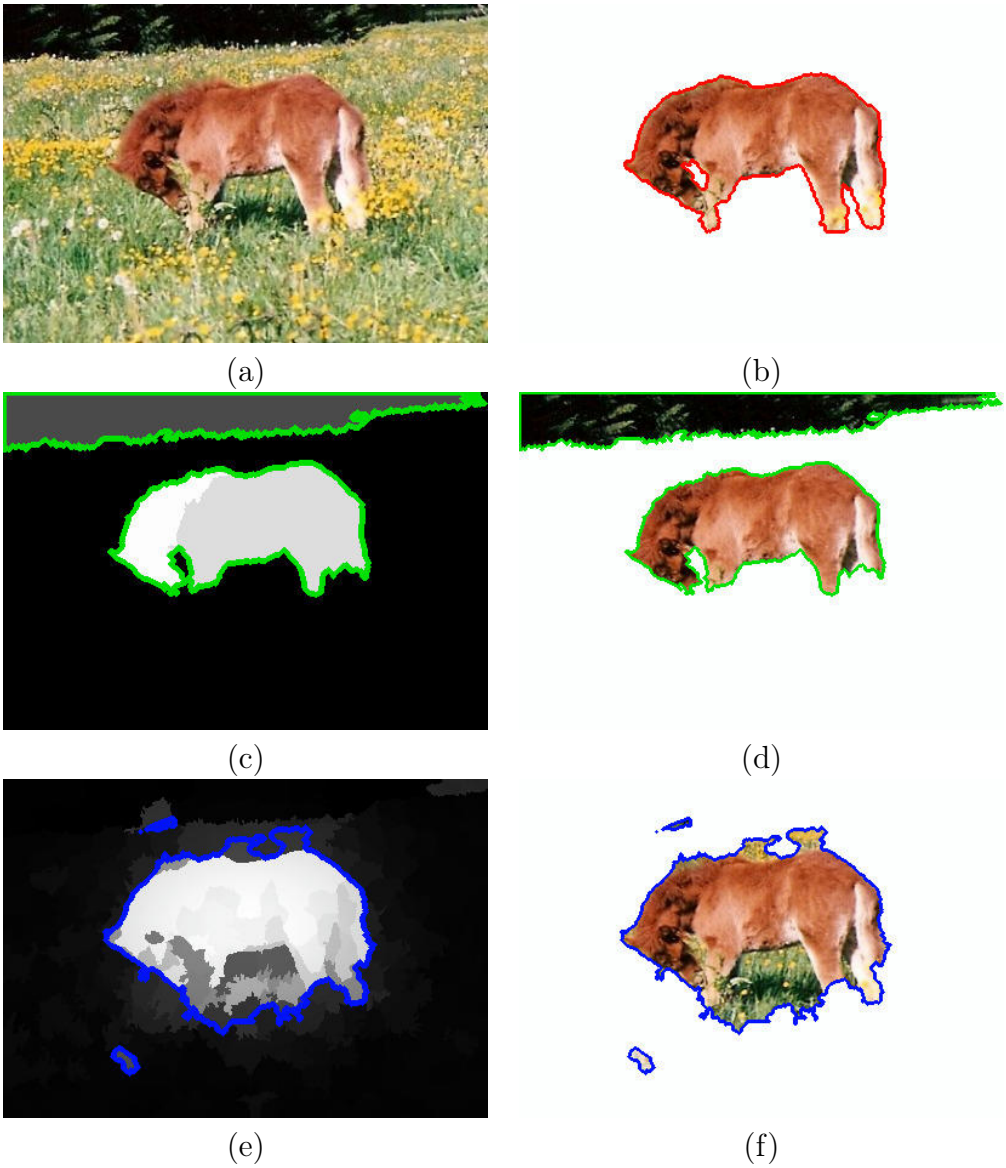


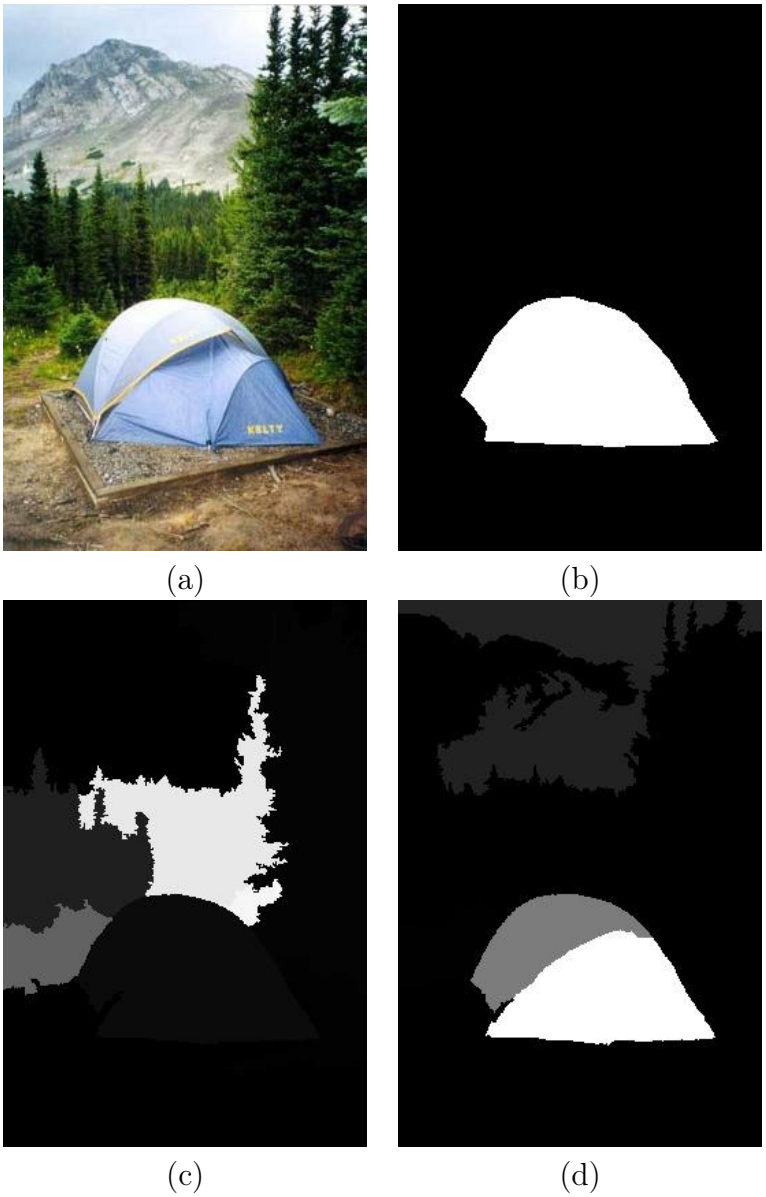
Figure 5.2: (a) Input image. (b) Ground-truth segmentation; (c-f) ITSELF/SMD saliency maps with mean-saliency threshold segmentation boundaries depicted on green/blue, respectively.

5.5 Non-natural-image dataset comparisons

Concerning the two evaluated non-natural-image datasets, ITSELF outperformed both SMD and DRFI by a big margin, especially on the parasite dataset (Table 5.2). Figure 5.9 shows side-by-side example results of the three methods. While DRFI and SMD create over-salient regions, ITSELF provides a better definition of the objects.

On the x-ray images, the inside of the lungs has different characteristics compared to the ribs. Because most of the patients' lung boundaries overlap with their ribs, ITSELF's results were not able to achieve a high BR score. However, ITSELF obtained substantially higher precision when compared to the other methods.

ITSELF fails to accurately detect both lungs on images where one of them is too small or has higher intensities when compared to the other (Figure 5.6). These are



(a) (b)

(c) (d)

Figure 5.3: (a) Original image. (b) Superpixel Segmentation; (c) Reported ITSELF result; (d) Improved result by removing the center and focus priors.

characteristics of not healthy images, so future works might use ITSELF segmentation error to indicate unhealthy patients.

A similar issue to the ribs happens on the parasite-eggs dataset. The parasite-eggs are enclosed by a membrane that often gets less colored than the egg's core (Figure 5.7).

ITSELF fails to accurately detect the salient parasite eggs on a few images where the impurities are elliptical, share similar colors, and are within the size range (Figure 5.8). Impurities too similar to the eggs are not present on most images, so ITSELF's BR and REC^{ω} scores are mostly affected by the miss-estimation of the membrane saliency.

(a) (b) (c)

Figure 5.4: (a) Original image. (b) Ground-truth; (c) ITSELF saliency map. On the second image, note how there are contrasting green parts on the image background.

5.6 Failed Attempts and Implementation Details

In this section, we will go over tests that did not provide considerable improvement compared to the methods described in this work.

We experimented using the mean color of the superpixels instead of computing the distance of quantized colors. By doing so, we sacrifice information to considerably reduce the number of operations to define the edge weights: Instead of using Equation 4.1, contrast would be simplified to $\boldsymbol{e}'(S,R) = \boldsymbol{e}(S,R)\exp^{\frac{\|c_S,c_R\|}{\sigma_s}}$, where $c_S, c_R \in C_I$ represent the mean color representing the superpixels S and R. By using a large number of regular superpixels, the information lost is less impactful. However, as discussed previously, superpixel regularity impacts boundary adherence, which results in worst object representation. However, we do keep the superpixel simplification strategy as a possibility for scenarios that require a larger number of superpixels.

We also tested computing one background/foreground map for each query superpixel to combine them later. Multiple query maps increase the number of computations and would only be acceptable if there were substantial improvements. The experiments did not result in any consistent improvement, but the processing time was increased considerably.

At an early stage of the framework, we implemented a GPU friendly version of the saliency estimation algorithm (Section 4.2). At the time, the adequate number of superpixels and colors used to achieve the best results was not high enough for the GPU version to outperform the optimized CPU version. Because the GPU implementation was a lot more restrictive and harder to tinker with, we only discontinued it. If today's implementation gets two slow when extending the framework to volumetric images, we will re-implement a GPU version of the current framework.

Regarding implementation details, we optimized some steps of the framework by creating look-up tables, allowing multi-threading, and storing information recurrent on multiple steps of the framework. Starting by the color distances and graph adjacency, we pre-compute the distances of all adjacent colors (*i.e.* all pairs of colors present in ad-

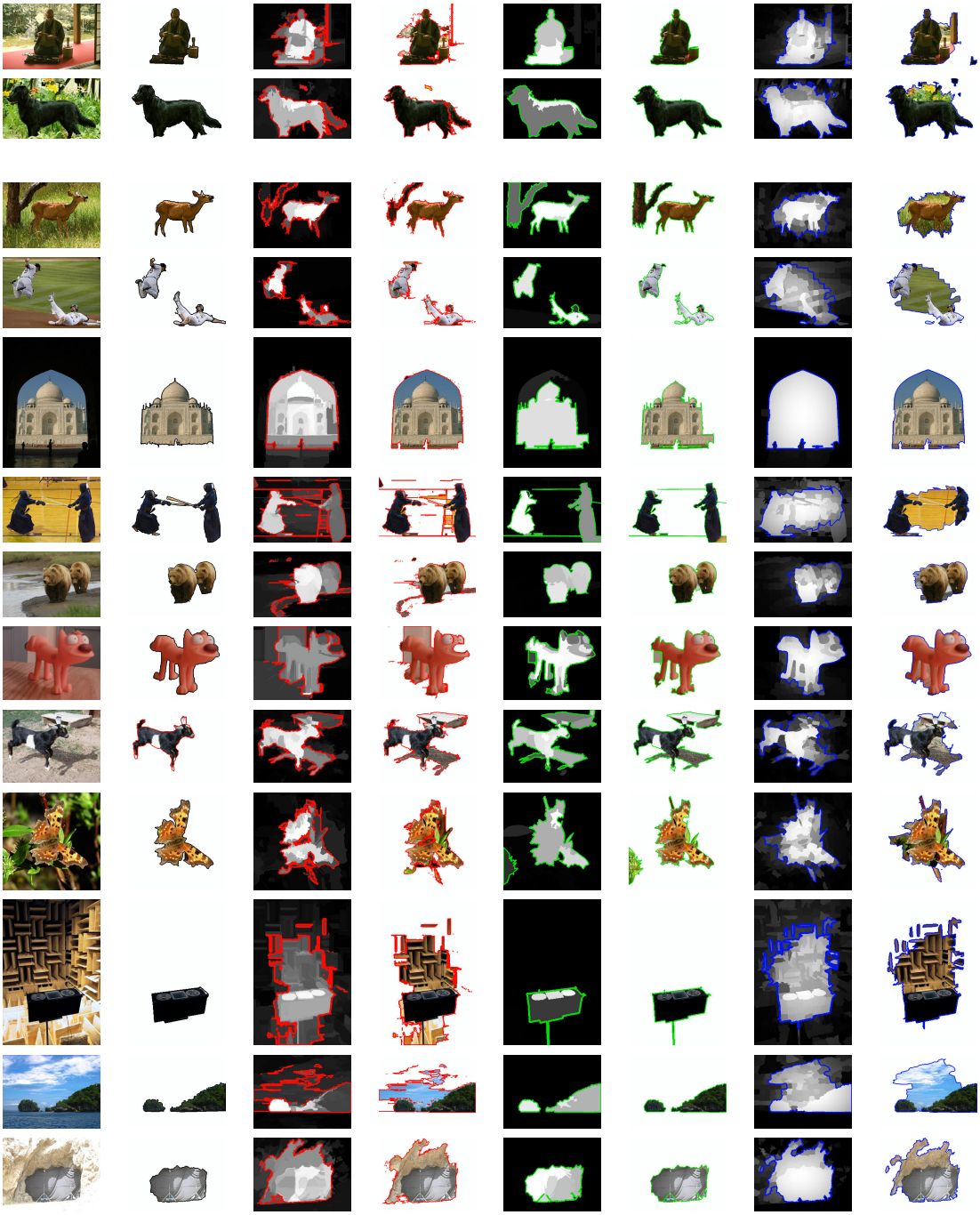
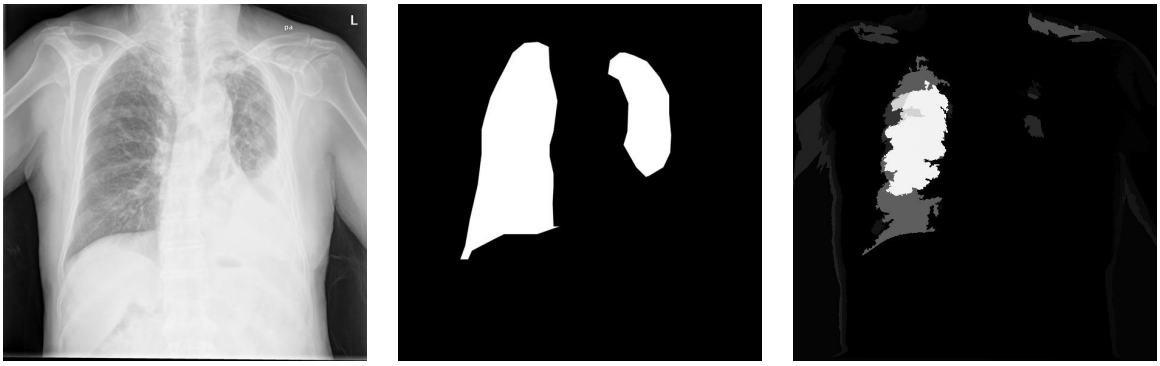


Figure 5.5: (a) Input image. (b) Ground-truth segmentation; (c-h) DRFI/ITSELF/SMD saliency maps with mean-saliency threshold segmentation boundaries depicted on red/green/blue, respectively. Note how ITSELF tend to create more accentuated contrast between the object and background, adhering to the boundaries.

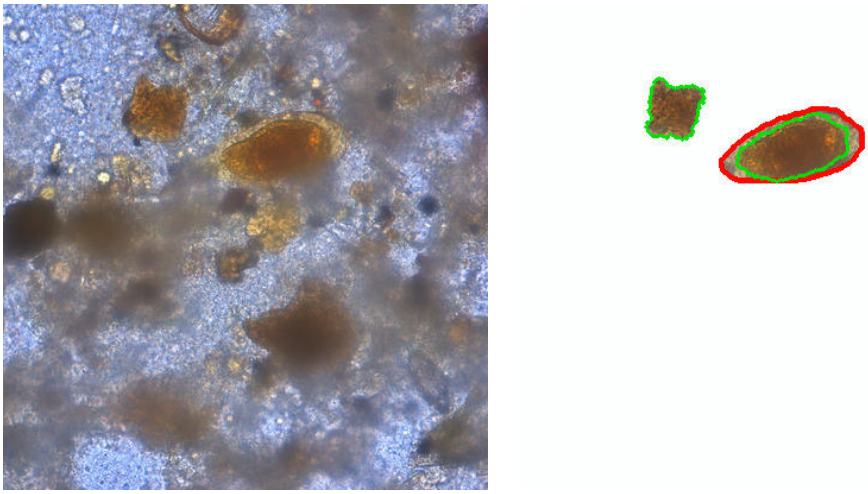
jacent superpixels) and store them on a look-up table. Using the color distance table considerably improved the computation time of Equation 4.1. Also, whenever the graph adjacency is changed (new superpixel segmentation, new query superpixels are selected), we update the color distance table, adding any pair of colors that are now adjacent.

The edge weight (Equation 4.1) and vertex saliency scores (Equation 4.2) are computed



(a) (b) (c)

Figure 5.6: (a) Original image. (b) Ground-truth; (c) ITSELF saliency map. ITSELF completely lost the smaller and brighter lung.



(a) (b)

Figure 5.7: (a) Original image. (b) Ground-truth (red) and ITSELF (green) segmentations overlayed. Note the lighter yellow membrane segmented on the ground-truth that was lost by ITSELF.

(a) (b) (c)

Figure 5.8: (a) Original image. (b) Ground-truth; (c) ITSELF saliency map. ITSELF highlights the top right impurity instead of the parasite-egg.

using multi-threading. Also, if the query importance $\psi' = 1$, we only consider query-based edges, and similarly, if $\psi' = 0$ we discard the query-based edges.

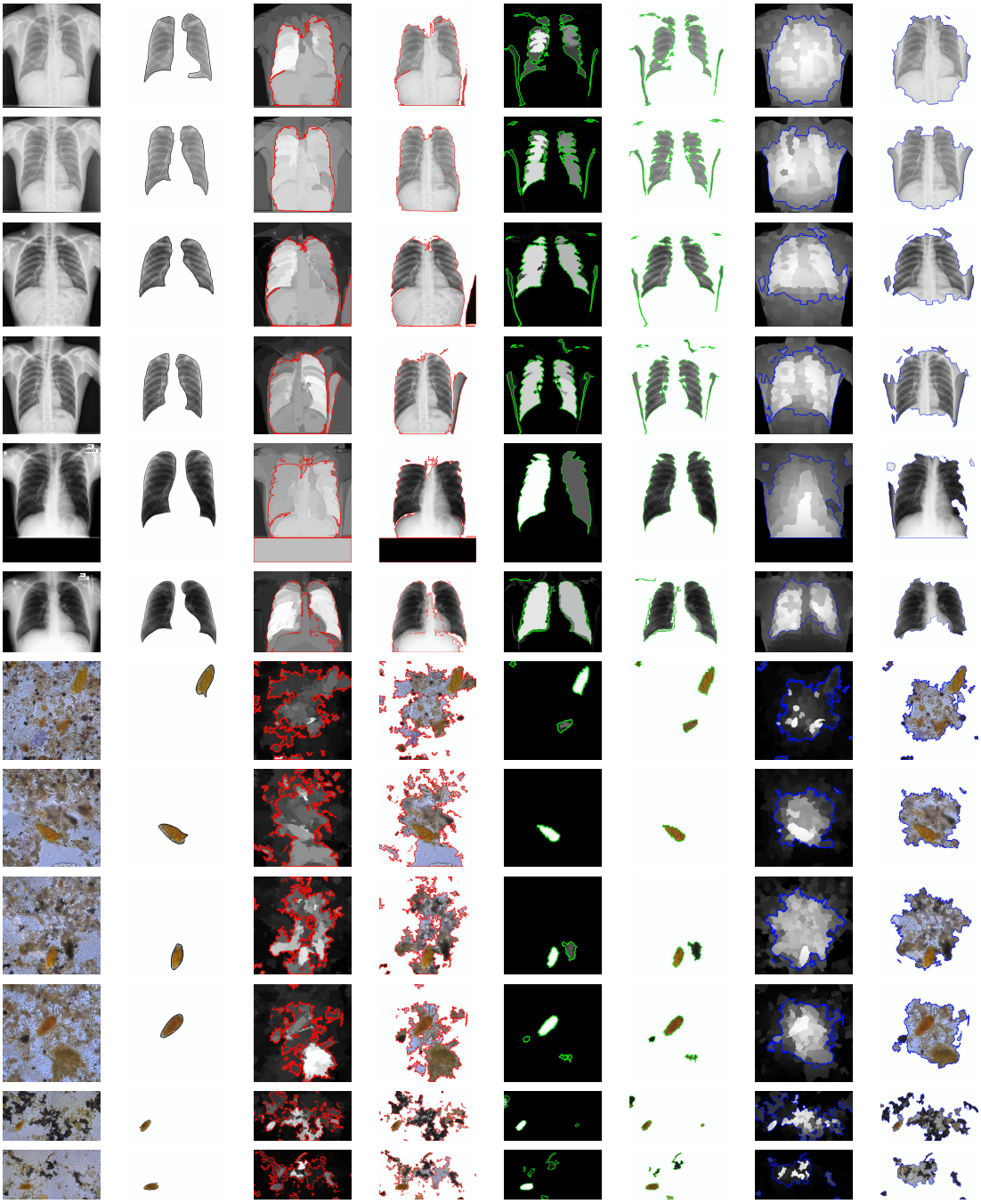


Figure 5.9: (a) Input image. (b) Ground-truth segmentation; (c-h) DRFI/ITSELF/SMD saliency maps with mean-saliency threshold segmentation boundaries depicted on red/green/blue, respectively Note how ITSELF tend to create more accentuated contrast between the object and background, adhezring to the boundaries.

	Methods	WF-Measure	BR	MAE	PRE^{ω}	REC^{ω}
ECSSD	**ITSELF**	0.509	0.473	0.177	0.601	0.491
	SMD	0.517	0.425	0.227	0.543	0.660
	DRFI	0.547	0.556	0.159	0.606	0.564
DUT	Methods	WF-Measure	BR	MAE	PRE^{ω}	REC^{ω}
	ITSELF	0.406	0.436	0.144	0.416	0.540
	SMD	0.424	0.353	0.136	0.387	0.659
	DRFI	0.357	0.356	0.193	0.290	0.679
ICOSEG	Methods	WF-Measure	BR	MAE	PRE^{ω}	REC^{ω}
	ITSELF	0.580	0.571	0.149	0.676	0.618
	SMD	0.611	0.527	0.138	0.696	0.656
	DRFI	0.547	0.591	0.152	0.657	0.635
MSRA10K	Methods	WF-Measure	BR	MAE	PRE^{ω}	REC^{ω}
	ITSELF	0.675	0.634	0.116	0.724	0.680
	SMD	0.704	0.594	0.104	0.730	0.733
	DRFI	0.583	0.435	0.149	0.525	0.724
Lungs	Methods	WF-Measure	BR	MAE	PRE^{ω}	REC^{ω}
	ITSELF	0.621	0.208	0.141	0.857	0.506
	SMD	0.404	0.095	0.325	0.294	0.724
	DRFI	0.325	0.255	0.412	0.224	0.664
Parasites	Methods	WF-Measure	BR	MAE	PRE^{ω}	REC^{ω}
	ITSELF	0.538	0.382	0.013	0.490	0.683
	SMD	0.121	0.192	0.155	0.078	0.662
	DRFI	0.041	0.320	0.164	0.022	0.433

Table 5.2: The best scores are colored in green and blue, respectively.

Chapter 6

Conclusion and Future work

We have presented ITSELF, a saliency estimation framework that is flexible for multiple image domains and allows the user to tailor salient characteristics as required. By using object-based superpixels, we proposed a novel loop interaction between saliency estimation and superpixel segmentation that iteratively improves both results. Thanks to that interaction, our method creates more semantically explainable maps and segmentation.

We compared our framework's implementations to two state-of-the-art saliency methods on six datasets, four of which are composed of natural images and two non-natural ones. We achieve competitive results on the natural images and outperformed by a significant margin on non-natural images. We provided possible ITSELF implementations, but we do not claim they are the optimal method for these datasets. Instead, the goal is to demonstrate the framework's flexibility to different scenarios.

It is also important to point out that the quality of a saliency estimator is only partially captured by the available metrics. In the current state of the literature, the task of salient object detection is more closely related to a soft object segmentation than to saliency estimation. The metrics often define a perfect saliency map to be the segmentation of the most salient objects of an image, with the ground-truth being the segmentation of the scene's object to be considered salient. Even with the more recent attempts to improve the metrics, the usage of binary segmentation as the ground-truth for saliency detection tasks imposes a limit to the quality of the metrics.

Regarding ITSELF improvements, the presented implementations use color and intensities as the main feature. However, other features may be adequate in different scenarios, as already demonstrated by the improvement provided by the feature selection step in *DRFI* [26]. In future implementations of the framework, we intend to explore other features such as texture, either through simple filters similar to DRFI or even deep-features provided by a neural network.

Another aspect yet to explore is the easy incorporation of other saliency methods, similar to the one presented in the cellular automata map integration [41]. The output of a saliency estimator can be used inside ITSELF with multiple functions: it can be used as a saliency map to be combined to the resulting iterations; it can be used to select query regions; and, it can be taken as a saliency-prior model. Due to ITSELF using a superpixel algorithm with high boundary adherence, it may be suitable for combining it to estimators that provide better detection even if they lack good delineation.

We also intend to further explore the usage of user-provided scribbles to model priors and queries. On the experiments performed, the scribble-based priors and queries were handled the same way as the domain-specific assumptions. However, the uncertainty of the assumptions force the usage of error-reducing strategies that should not be taken into consideration when using user supervision. Therefore, we intend to improve high confidence priors and queries by creating better fit strategies for them.

Lastly, we want to extend ITSELF for volumetric images. The algorithm does not require many changes apart from the map integration step that will require stacking the images into a four-dimensional grid and defining an adjacency relation in it. The major challenge overall will be the algorithm's performance due to the large volume of data.

In conclusion, we proposed an easily extensible saliency estimation framework that has shown the potential to perform well on multiple image domains and adequately estimate the saliency of diverse objects. However, several experiments, extensions, and combinations are to be tried and implemented within ITSELF to make it viable to other applications.

Bibliography

[1] Radhakrishna Achanta, Francisco Estrada, Patricia Wils, and Sabine Süsstrunk. Salient region detection and segmentation. In *International conference on computer vision systems*, pages 66–75. Springer, 2008.

[2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[3] Eduardo Barreto Alexandre, Ananda Shankar Chowdhury, Alexandre Xavier Falcao, and Paulo A Vechiatto Miranda. Ift-slic: A general framework for superpixel generation based on simple linear iterative clustering and image foresting transform. In *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 337–344. IEEE, 2015.

[4] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. Interactively co-segmentating topically related images with intelligent scribble guidance. *International journal of computer vision*, 93(3):273–292, 2011.

[5] Felipe Belém, Silvio Jamil Guimarães, and Alexandre Xavier Falcão. Superpixel segmentation by object-based iterative spanning forest. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 334–341. Springer International Publishing, 2019.

[6] Felipe Belém, Leonardo Melo, Silvio Guimarães, and Alexandre Falcão. The importance of object-based seed sampling for superpixel segmentation. In *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 108–115. IEEE, 2019.

[7] Felipe C Belém, Silvio Jamil F Guimarães, and Alexandre X Falcao. Superpixel segmentation using dynamic and iterative spanning forest. *IEEE Signal Processing Letters*, 27:1440–1444, 2020.

[8] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection: A survey. *Computational visual media*, pages 1–34, 2019.

[9] Sema Candemir, Stefan Jaeger, Kannappan Palaniappan, Jonathan P Musco, Rahul K Singh, Zhiyun Xue, Alexandros Karargyris, Sameer Antani, George Thoma,

and Clement J McDonald. Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration. *IEEE transactions on medical imaging*, 33(2):577–590, 2013.

[10] César Castelo-Fernández and Alexandre X Falcão. Learning visual dictionaries from class-specific superpixel segmentation. In *International Conference on Computer Analysis of Images and Patterns*, pages 171–182. Springer, 2019.

[11] Shuhan Chen, Xiuli Tan, Ben Wang, and Xuelong Hu. Reverse attention for salient object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 234–250, 2018.

[12] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. Sketch2photo: Internet image montage. In *ACM transactions on graphics (TOG)*, volume 28, page 124. ACM, 2009.

[13] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, 2014.

[14] Krzysztof Chris Ciesielski, Alexandre Xavier Falcão, and Paulo AV Miranda. Path-value functions for which dijkstra's algorithm returns optimal mapping. *Journal of Mathematical Imaging and Vision*, 60(7):1025–1036, 2018.

[15] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[17] Alexandre Falcão and Jordão Bragantini. The role of optimum connectivity in image segmentation: Can the algorithm learn object information during the process? In *International Conference on Discrete Geometry for Computer Imagery*, pages 180–194. Springer, 2019.

[18] Alexandre X Falcão, Jorge Stolfi, and Roberto de Alencar Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 26(1):19–29, 2004.

[19] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.

[20] Felipe Lemes Galvão, Alexandre Xavier Falcão, and Ananda Shankar Chowdhury. Risf: Recursive iterative spanning forest for superpixel segmentation. In *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 408–415. IEEE, 2018.

[21] Jancarlo Ferreira Gomes, Sumie Hoshino-Shimizu, Luiz Cândido S. Dias, Ana Julia SA Araujo, Vera LP Castilho, and Fatima AMA Neves. Evaluation of a novel kit (tf-test) for the diagnosis of intestinal parasitic infections. *Journal of clinical laboratory analysis*, 18(2):132–138, 2004.

[22] Chenlei Guo and Liming Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE transactions on image processing*, 19(1):185–198, 2009.

[23] Shengfeng He, Rynson W.H. Lau, Wenxi Liu, Zhe Huang, and Qingxiong Yang. Supercnn: A superpixelwise convolutional neural network for salient object detection. *International journal of computer vision*, 115(3):330–344, 2015.

[24] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.

[25] Stefan Jaeger, Alexandros Karargyris, Sema Candemir, Les Folio, Jenifer Siegelman, Fiona Callaghan, Zhiyun Xue, Kannappan Palaniappan, Rahul K Singh, Sameer Antani, et al. Automatic tuberculosis screening using chest radiographs. *IEEE transactions on medical imaging*, 33(2):233–245, 2013.

[26] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2083–2090, 2013.

[27] Peng Jiang, Haibin Ling, Jingyi Yu, and Jingliang Peng. Salient region detection by ufo: Uniqueness, focusness and objectness. In *Proceedings of the IEEE international conference on computer vision*, pages 1976–1983, 2013.

[28] Gayoung Lee, Yu-Wing Tai, and Junmo Kim. Deep saliency with encoded low level distance map and high level features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 660–668, 2016.

[29] Thomas Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International journal of computer vision*, 43(1):29–44, 2001.

[30] Bo Li, Zhengxing Sun, and Yuqi Guo. Supervae: Superpixelwise variational autoencoder for salient object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8569–8576, 2019.

[31] Xiao Lin, Zhi-Jie Wang, Lizhuang Ma, and Xiabao Wu. Saliency detection via multi-scale global cues. *IEEE Transactions on Multimedia*, 21(7):1646–1659, 2018.

[32] Hantao Liu and Ingrid Heynderickx. Visual attention in objective image quality assessment: Based on eye-tracking data. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(7):971–982, 2011.

[33] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE Transactions on Pattern analysis and machine intelligence*, 33(2):353–367, 2010.

[34] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[35] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 248–255, 2014.

[36] Samuel B Martins, Guilherme Ruppert, Fabiano Reis, Clarissa L Yasuda, and Alexandre X Falcão. A supervoxel-based approach for unsupervised abnormal asymmetry detection in mr images of the brain. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 882–885. IEEE, 2019.

[37] Paulo AV Miranda, R da S Torres, and Alexandre X Falcao. Tsd: a shape descriptor based on a distribution of tensor scale local orientation. In *XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05)*, pages 139–146. IEEE, 2005.

[38] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[39] Houwen Peng, Bing Li, Haibin Ling, Weiming Hu, Weihua Xiong, and Stephen J Maybank. Salient object detection via structured matrix decomposition. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):818–832, 2016.

[40] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7479–7489, 2019.

[41] Yao Qin, Huchuan Lu, Yiqun Xu, and He Wang. Saliency detection via cellular automata. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 110–119, 2015.

[42] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. In *Advances in neural information processing systems*, pages 3347–3357, 2019.

[43] Leonardo Marques Rocha, Fábio AM Cappabianco, and Alexandre Xavier Falcão. Data clustering as an optimum-path forest problem with applications in image analysis. *International Journal of Imaging Systems and Technology*, 19(2):50–68, 2009.

[44] Xiaohui Shen and Ying Wu. A unified approach to salient object detection via low rank matrix recovery. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 853–860. IEEE, 2012.

[45] Meng Tang, Lena Gorelick, Olga Veksler, and Yuri Boykov. Grabcut in one cut. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1769–1776, 2013.

[46] Na Tong, Huchuan Lu, Lihe Zhang, and Xiang Ruan. Saliency detection with multi-scale superpixels. *IEEE Signal Processing Letters*, 21(9):1035–1039, 2014.

[47] Roberto Valenti, Nicu Sebe, and Theo Gevers. Image saliency by isocentric curvedness and color. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2185–2192. IEEE, 2009.

[48] John E Vargas-Muñoz, Ananda S Chowdhury, Eduardo B Alexandre, Felipe L Galvão, Paulo A Vechiatto Miranda, and Alexandre X Falcão. An iterative spanning forest framework for superpixel segmentation. *IEEE Transactions on Image Processing*, 28(7):3477–3489, 2019.

[49] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Deep networks for saliency detection via local estimation and global search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3183–3192, 2015.

[50] Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, Haibin Ling, and Ruigang Yang. Salient object detection in the deep learning era: An in-depth survey. *arXiv preprint arXiv:1904.09146*, 2019.

[51] Yunchao Wei, Xiaodan Liang, Yunpeng Chen, Xiaohui Shen, Ming-Ming Cheng, Jiashi Feng, Yao Zhao, and Shuicheng Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2314–2320, 2016.

[52] Xiyin Wu, Xiaodi Ma, Jinxia Zhang, and Zhong Jin. Salient object detection via reliable boundary seeds and saliency refinement. *IET Computer Vision*, 13(3):302–311, 2018.

[53] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1155–1162, 2013.

[54] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3166–3173, 2013.

[55] Wenxian Yang, Jianfei Cai, Jianmin Zheng, and Jiebo Luo. User-friendly interactive image segmentation through unified combinatorial user inputs. *IEEE Transactions on Image Processing*, 19(9):2470–2479, 2010.

[56] Yi Zeng, Pingping Zhang, Jianming Zhang, Zhe Lin, and Huchuan Lu. Towards high-resolution salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7234–7243, 2019.

[57] Dingwen Zhang, Junwei Han, and Yu Zhang. Supervision by fusion: Towards unsupervised learning of deep salient object detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4048–4056, 2017.

[58] Jing Zhang, Tong Zhang, Yuchao Dai, Mehrtash Harandi, and Richard Hartley. Deep unsupervised saliency detection: A multiple noisy labeling perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9029–9038, 2018.

[59] Jinxia Zhang, Shixiong Fang, Krista A Ehinger, Haikun Wei, Wankou Yang, Kanjian Zhang, and Jingyu Yang. Hypergraph optimization for salient region detection based on foreground and background queries. *IEEE Access*, 6:26729–26741, 2018.

[60] Lihe Zhang, Chuan Yang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Ranking saliency. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1892–1904, 2016.

[61] Xue Zhang, Zheng Wang, Qinghua Hu, Jinchang Ren, and Meijun Sun. Boundary-aware high-resolution network with region enhancement for salient object detection. *Neurocomputing*, 418:91–101, 2020.

[62] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1265–1274, 2015.

[63] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2814–2821, 2014.